(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 September 2003 (04.09.2003)

PCT

(10) International Publication Number
WO 03/073770 A1

(51) International Patent Classification[7]: H04N 7/24

(21) International Application Number: PCT/US03/05636

(22) International Filing Date: 24 February 2003 (24.02.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/359,606	25 February 2002 (25.02.2002)	US
60/361,773	5 March 2002 (05.03.2002)	US
60/363,643	8 March 2002 (08.03.2002)	US
10/371,438	21 February 2003 (21.02.2003)	US

(71) Applicant: **SONY ELECTRONICS, INC.** [US/US]; 1 Sony Drive, Park Ridge, NJ 07656 (US).

(72) Inventors: **VISHARAM, Mohammed, Zubair**; 444 Saratoga Avenue, #26A, Santa Clara, CA 95050 (US). **TABATABAI, Ali**; 10265 E. Estates Drive, Cupertino, CA 95014 (US). **WALKER, Toby**; 604 Galer Street, #321, Seattle, WA 98109 (US).

(74) Agents: **SALTER, James, H.** et al.; Blakely, Sokoloff, Taylor & Zafman, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, CA 90025-1026 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

RCV AA PW030288
CITED BY APPLICANT

WO 03/073770 A1

(54) Title: METHOD AND APPARATUS FOR SUPPORTING AVC IN MP4

(57) Abstract: Paramater set metadata identifying parameter sets for multiple portions of multimedia data is created. Further, a file associated with the multimedia data is formed. This file includes the parameter set metadata, as well as other information pertaining to the multimedia data.

METHOD AND APPARATUS FOR SUPPORTING AVC IN MP4

RELATED APPLICATIONS

This application is related to and claims the benefit of U.S. Provisional Patent applications serial numbers 60/359,606 filed February 25, 2002, 60/361,773, filed March 5, 2002, and 60/363,643, filed March 8, 2002, which are hereby incorporated by reference.

FIELD OF THE INVENTION

The invention relates generally to the storage and retrieval of audiovisual content in a multimedia file format and particularly to file formats compatible with the ISO media file format.

COPYRIGHT NOTICE/PERMISSION

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 2001, Sony Electronics, Inc., All Rights Reserved.

BACKGROUND OF THE INVENTION

In the wake of rapidly increasing demand for network, multimedia, database and other digital capacity, many multimedia coding and storage schemes have evolved. One of the well known file formats for encoding and storing audiovisual data is the QuickTime® file format developed by Apple Computer Inc. The QuickTime file format was used as the starting point for creating the International Organization for Standardization (ISO) Multimedia file format, ISO/IEC 14496-12, Information Technology – Coding of audio-visual objects – Part 12: ISO Media File Format (also known as the ISO file format), which was, in turn, used as a template for two standard file formats: (1) For an MPEG-4 file format developed by the Moving Picture Experts Group, known as MP4 (ISO/IEC 14496-14, Information Technology -- Coding of audio-visual

objects -- Part 14: MP4 File Format); and (2) a file format for JPEG 2000 (ISO/IEC 15444-1), developed by Joint Photographic Experts Group (JPEG).

The ISO media file format is composed of object-oriented structures referred to as boxes (also referred to as atoms or objects). The two important top-level boxes contain either media data or metadata. Most boxes describe a hierarchy of metadata providing declarative, structural and temporal information about the actual media data. This collection of boxes is contained in a box known as the movie box. The media data itself may be located in media data boxes or externally. Each media data stream is called a track (also known as an elementary stream or simply a stream).

The primary metadata is the movie object. The movie box includes track boxes, which describe temporally presented media data. The media data for a track can be of various types (e.g., video data, audio data, binary format screen representations (BIFS), etc.). Each track is further divided into samples (also known as access units or pictures). A sample represents a unit of media data at a particular time point. Sample metadata is contained in a set of sample boxes. Each track box contains a sample table box metadata box, which contains boxes that provide the time for each sample, its size in bytes, and its location (external or internal to the file) for its media data, and so forth. A sample is the smallest data entity which can represent timing, location, and other metadata information.

Recently, MPEG's video group and Video Coding Experts Group (VCEG) of International Telecommunication Union (ITU) began working together as a Joint Video Team (JVT) to develop a new video coding/decoding (codec) standard referred to as ITU Recommendation H.264 or MPEG-4-Part 10, Advanced Video Codec (AVC) or JVT codec. These terms, and their abbreviations such as H.264, JVT, and AVC are used interchangeably here.

The JVT codec design distinguished between two different conceptual layers, the Video Coding Layer (VCL), and the Network Abstraction Layer (NAL). The VCL contains the coding related parts of the codec, such as motion compensation, transform coding of coefficients, and entropy coding. The output of the VCL is slices, each of which contains a series of macroblocks and associated header information. The NAL abstracts the VCL from the details of the transport layer used to carry the VCL data. It defines a generic and transport independent representation for information above the level of the slice. The NAL defines the interface between the video codec itself and the outside world. Internally, the NAL uses NAL packets. A NAL packet includes a type field indicating the

type of the payload plus a set of bits in the payload. The data within a single slice can be divided further into different data partitions.

In many existing video coding formats, the coded stream data includes various kinds of headers containing parameters that control the decoding process. For example, the MPEG-2 video standard includes sequence headers, enhanced group of pictures (GOP), and picture headers before the video data corresponding to those items. In JVT, the information needed to decode VCL data is grouped into parameter sets. Each parameter set is given an identifier that is subsequently used as a reference from a slice. Instead of sending the parameter sets inside (in-band) the stream, they can be sent outside (out-of-band) the stream.

Existing file formats do not provide a facility for storing the parameter sets associated with coded media data; nor do they provide a means for efficiently linking media data (i.e., samples or sub-samples) to parameters sets so that parameter sets can be efficiently retrieved and transmitted.

In the ISO media file format, the smallest unit that can be accessed without parsing media data is a sample, i.e., a whole picture in AVC. In many coded formats, a sample can be further divided into smaller units called sub-samples (also referred to as sample fragments or access unit fragments). In the case of AVC, a sub-sample corresponds to a slice. However, existing file formats do not support accessing sub-parts of a sample. For systems that need to flexibly form data stored in a file into packets for streaming, this lack of access to sub-samples hinders flexible packetization of JVT media data for streaming.

Another limitation of existing storage formats has to do with switching between stored streams with different bandwidth in response to changing network conditions when streaming media data. In a typical streaming scenario, one of the key requirements is to scale the bit rate of the compressed data in response to changing network conditions. This is typically achieved by encoding multiple streams with different bandwidth and quality settings for representative network conditions and storing them in one or more files. The server can then switch among these pre-coded streams in response to network conditions. In existing file formats, switching between streams is only possible at samples that do not depend on prior samples for reconstruction. Such samples are referred to as I-frames. No support is currently provided for switching between streams at samples that depend on prior samples for reconstruction (i.e., a P-frame or a B-frame that depend on multiple samples for reference).

The AVC standard provides a tool known as switching pictures (called SI- and SP-pictures) to enable efficient switching between streams, random access, and error resilience, as well as other features. A switching picture is a special type of picture whose reconstructed value is exactly equivalent to the picture it is supposed to switch to. Switching pictures can use reference pictures differing from those used to predict the picture that they match, thus providing more efficient coding than using I-frames. To use switching pictures stored in a file efficiently it is necessary to know which sets of pictures are equivalent and to know which pictures are used for prediction. Existing file formats do not provide this information and therefore this information must be extracted by parsing the coded stream, which is inefficient and slow.

Thus, there is a need to enhance storage methods to address the new capabilities provided by emerging video coding standards and to address the existing limitations of those storage methods.

SUMMARY OF THE INVENTION

Parameter set metadata identifying parameter sets for multiple portions of multimedia data is created. Further, a file associated with the multimedia data is formed. This file includes the parameter set metadata, as well as other information pertaining to the multimedia data.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

Figure 1 is a block diagram of one embodiment of an encoding system;

Figure 2 is a block diagram of one embodiment of a decoding system;

Figure 3 is a block diagram of a computer environment suitable for practicing the invention;0

Figure 4 is a flow diagram of a method for storing sub-sample metadata at an encoding system;

Figure 5 is a flow diagram of a method for utilizing sub-sample metadata at a decoding system;

Figure 6 illustrates an extended MP4 media stream model with sub-samples;

Figures 7A – 7K illustrate exemplary data structures for storing sub-sample metadata;

Figure 8 is a flow diagram of a method for storing parameter set metadata at an encoding system;

Figure 9 is a flow diagram of a method for utilizing parameter set metadata at a decoding system;

Figures 10A – 10E illustrate exemplary data structures for storing parameter set metadata;

Figure 11 illustrates an exemplary enhanced group of pictures (GOP);

Figure 12 is a flow diagram of a method for storing sequences metadata at an encoding system;

Figure 13 is a flow diagram of a method for utilizing sequences metadata at a decoding system;

Figures 14A – 14E illustrate exemplary data structures for storing sequences metadata;

Figures 15A and 15B illustrate the use of a switch sample set for bit stream switching;

Figure 15C is a flow diagram of one embodiment of a method for determining a point at which a switch between two bit streams is to be performed;

Figure 16 is a flow diagram of a method for storing switch sample metadata at an encoding system;

Figure 17 is a flow diagram of a method for utilizing switch sample metadata at a decoding system;

Figure 18 illustrates an exemplary data structure for storing switch sample metadata;

Figures 19A and 19B illustrate the use of a switch sample set to facilitate random access entry points into a bit stream;

Figure 19C is a flow diagram of one embodiment of a method for determining a random access point for a sample;

Figures 20A and 20B illustrate the use of a switch sample set to facilitate error recovery; and

Figure 20C is a flow diagram of one embodiment of a method for facilitating error recovery when sending a sample.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references indicate similar elements, and in which is shown, by way of illustration, specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical, functional and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

Overview

Beginning with an overview of the operation of the invention, **Figure 1** illustrates one embodiment of an encoding system 100. The encoding system 100 includes a media encoder 104, a metadata generator 106 and a file creator 108. The media encoder 104 receives media data that may include video data (e.g., video objects created from a natural source video scene and other external video objects), audio data (e.g., audio objects created from a natural source audio scene and other external audio objects), synthetic objects, or any combination of the above. The media encoder 104 may consist of a number of individual encoders or include sub-encoders to process various types of media data. The media encoder 104 codes the media data and passes it to the metadata generator 106. The metadata generator 106 generates metadata that provides information about the media data according to a media file format. The media file format may be derived from the ISO media file format (or any of its derivatives such as MPEG-4, JPEG 2000, etc.), QuickTime or any other media file format, and also include some additional data structures. In one embodiment, additional data structures are defined to store metadata pertaining to sub-samples within the media data. In another embodiment, additional data structures are defined to store metadata linking portions of media data (e.g., samples or sub-samples) to corresponding parameter sets which include decoding information that has been traditionally stored in the media data. In yet another embodiment, additional data structures are defined to store metadata pertaining to various groups of samples within the metadata that are created based on inter-dependencies of the samples in the media data. In

still another embodiment, an additional data structure is defined to store metadata pertaining to switch sample sets associated with the media data. A switch sample set refers to a set of samples that have identical decoding values but may depend on different samples. In yet other embodiments, various combinations of the additional data structures are defined in the file format being used. These additional data structures and their functionality will be described in greater detail below.

The file creator 108 stores the metadata in a file whose structure is defined by the media file format. In one embodiment, the file contains both the coded media data and metadata pertaining to that media data. Alternatively, the coded media data is included partially or entirely in a separate file and is linked to the metadata by references contained in the metadata file (e.g., via URLs). The file created by the file creator 108 is available on a channel 110 for storage or transmission.

Figure 2 illustrates one embodiment of a decoding system 200. The decoding system 200 includes a metadata extractor 204, a media data stream processor 206, a media decoder 210, a compositor 212 and a renderer 214. The decoding system 200 may reside on a client device and be used for local playback. Alternatively, the decoding system 200 may be used for streaming data and have a server portion and a client portion communicating with each other over a network (e.g., Internet) 208. The server portion may include the metadata extractor 204 and the media data stream processor 206. The client portion may include the media decoder 210, the compositor 212 and the renderer 214.

The metadata extractor 204 is responsible for extracting metadata from a file stored in a database 216 or received over a network (e.g., from the encoding system 100). The file may or may not include media data associated with the metadata being extracted. The metadata extracted from the file includes one or more of the additional data structures described above.

The extracted metadata is passed to the media data stream processor 206 which also receives the associated coded media data. The media data stream processor 206 uses the metadata to form a media data stream to be sent to the media decoder 210. In one embodiment, the media data stream processor 206 uses metadata pertaining to sub-samples to locate sub-samples in the media data (e.g., for packetization). In another embodiment, the media data stream processor 206 uses metadata pertaining to parameter sets to link portions of the media data to its corresponding parameter sets. In yet another

embodiment, the media data stream processor 206 uses metadata defining various groups of samples within the metadata to access samples in a certain group (e.g., for scalability by dropping a group containing samples on which no other samples depend to lower the transmitted bit rate in response to transmission conditions). In still another embodiment, the media data stream processor 206 uses metadata defining switch sample sets to locate a switch sample that has the same decoding value as the sample it is supposed to switch to but does not depend on the samples on which this resultant sample would depend on (e.g., to allow switching to a stream with a different bit-rate at a P-frame or B-frame).

Once the media data stream is formed, it is sent to the media decoder 210 either directly (e.g., for local playback) or over a network 208 (e.g., for streaming data) for decoding. The compositor 212 receives the output of the media decoder 210 and composes a scene which is then rendered on a user display device by the renderer 214.

The following description of **Figure 3** is intended to provide an overview of computer hardware and other operating components suitable for implementing the invention, but is not intended to limit the applicable environments. **Figure 3** illustrates one embodiment of a computer system suitable for use as a metadata generator 106 and/or a file creator 108 of **Figure 1**, or a metadata extractor 204 and/or a media data stream processor 206 of **Figure 2**.

The computer system 340 includes a processor 350, memory 355 and input/output capability 360 coupled to a system bus 365. The memory 355 is configured to store instructions which, when executed by the processor 350, perform the methods described herein. Input/output 360 also encompasses various types of computer-readable media, including any type of storage device that is accessible by the processor 350. One of skill in the art will immediately recognize that the term "computer-readable medium/media" further encompasses a carrier wave that encodes a data signal. It will also be appreciated that the system 340 is controlled by operating system software executing in memory 355. Input/output and related media 360 store the computer-executable instructions for the operating system and methods of the present invention. Each of the metadata generator 106, the file creator 108, the metadata extractor 204 and the media data stream processor 206 that are shown in **Figures 1and 2** may be a separate component coupled to the processor 350, or may be embodied in computer-executable instructions executed by the processor 350. In one embodiment, the computer system 340 may be part of, or coupled to, an ISP (Internet Service Provider) through input/output 360 to transmit or receive

media data over the Internet. It is readily apparent that the present invention is not limited to Internet access and Internet web-based sites; directly coupled and private networks are also contemplated.

It will be appreciated that the computer system 340 is one example of many possible computer systems that have different architectures. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor. One of skill in the art will immediately appreciate that the invention can be practiced with other computer system configurations, including multiprocessor systems, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network.

Sub-Sample Accessibility

Figures 4 and 5 illustrate processes for storing and retrieving sub-sample metadata that are performed by the encoding system 100 and the decoding system 200 respectively. The processes may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both. For software-implemented processes, the description of a flow diagram enables one skilled in the art to develop such programs including instructions to carry out the processes on suitably configured computers (the processor of the computer executing the instructions from computer-readable media, including memory). The computer-executable instructions may be written in a computer programming language or may be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the embodiments of the present invention are not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or produce a result. It will be appreciated that more or fewer operations may be incorporated

into the processes illustrated in **Figures 4 and 5** without departing from the scope of the invention and that no particular order is implied by the arrangement of blocks shown and described herein.

Figure 4 is a flow diagram of one embodiment of a method 400 for creating sub-sample metadata at the encoding system 100. Initially, method 400 begins with processing logic receiving a file with encoded media data (processing block 402). Next, processing logic extracts information that identifies boundaries of sub-samples in the media data (processing block 404). Depending on the file format being used, the smallest unit of the data stream to which a time attribute can be attached is referred to as a sample (as defined by the ISO media file format or QuickTime), an access unit (as defined by MPEG-4), or a picture (as defined by JVT), etc. A sub-sample represents a contiguous portion of a data stream below the level of a sample. The definition of a sub-sample depends on the coding format but, in general, a sub-sample is a meaningful sub-unit of a sample that may be decoded as a singly entity or as a combination of sub-units to obtain a partial reconstruction of a sample. A sub-sample may also be called an access unit fragment. Often, sub-samples represent divisions of a sample's data stream so that each sub-sample has few or no dependencies on other sub-samples in the same sample. For example, in JVT, a sub-sample is a NAL packet. Similarly, for MPEG-4 video, a sub-sample would be a video packet.

In one embodiment, the encoding system 100 operates at the Network Abstraction Layer defined by JVT as described above. The JVT media data stream consists of a series of NAL packets where each NAL packet (also referred to as a NAL unit) contains a header part and a payload part. One type of NAL packet is used to include coded VCL data for each slice, or a single data partition of a slice. In addition, a NAL packet may be an information packet including supplemental enhancement information (SEI) messages. SEI messages represent optional data to be used in the decoding of corresponding slices. In JVT, a sub-sample could be a complete NAL packet with both header and payload.

At processing block 406, processing logic creates sub-sample metadata that defines sub-samples in the media data. In one embodiment, the sub-sample metadata is organized into a set of predefined data structures (e.g., a set of boxes). The set of predefined data structures may include a data structure containing information about the size of each sub-sample, a data structure containing information about the total number of sub-samples in each sample, a data structure containing information describing each sub-sample (e.g.,

what is defined as a sub-sample), or any other data structures containing data pertaining to the sub-samples.

Next, in one embodiment, processing logic determines whether any data structure contains a repeated sequence of data (decision box 408). If this determination is positive, processing logic converts each repeated sequence of data into a reference to a sequence occurrence and the number of times the repeated sequence occurs (processing block 410).

Afterwards, at processing block 412, processing logic includes the sub-sample metadata into a file associated with media data using a specific media file format (e.g., the JVT file format). Depending on the media file format, the sub-sample metadata may be stored with sample metadata (e.g., sub-sample data structures may be included in a sample table box containing sample data structures) or independently from the sample metadata.

Figure 5 is a flow diagram of one embodiment of a method 500 for utilizing sub-sample metadata at the decoding system 200. Initially, method 500 begins with processing logic receiving a file associated with encoded media data (processing block 502). The file may be received from a database (local or external), the encoding system 100, or from any other device on a network. The file includes sub-sample metadata that defines sub-samples in the media data.

Next, processing logic extracts the sub-sample metadata from the file (processing block 504). As discussed above, the sub-sample metadata may be stored in a set of data structures (e.g., a set of boxes).

Further, at processing block 506, processing logic uses the extracted metadata to identify sub-samples in the encoded media data (stored in the same file or in a different file) and combines various sub-samples into packets to be sent to a media decoder, thus enabling flexible packetization of media data for streaming (e.g., to support error resilience, scalability, etc.).

Exemplary sub-sample metadata structures will now be described with reference to an extended ISO media file format (referred to as an extended MP4). It will be obvious to one versed in the art that other media file formats could be easily extended to incorporate similar data structures for storing sub-sample metadata.

Figure 6 illustrates the extended MP4 media stream model with sub-samples. Presentation data (e.g., a presentation containing synchronized audio and video) is represented by a movie 602. The movie 602 includes a set of tracks 604. Each track 604 represents a media data stream. Each track 604 is divided into samples 606. Each sample

606 represents a unit of media data at a particular time point. A sample 606 is further divided into sub-samples 608. In the JVT standard, a sub-sample 608 may represent a NAL packet or unit, such as a single slice of a picture, one data partition of a slice with multiple data partitions, an in-band parameter set, or an SEI information packet. Alternatively, a sub-sample 606 may represent any other structured element of a sample, such as the coded data representing a spatial or temporal region in the media. In one embodiment, any partition of the coded media data according to some structural or semantic criterion can be treated as a sub-sample.

Figures 7A – 7L illustrate exemplary data structures for storing sub-sample metadata.

Referring to **Figure 7A**, a sample table box 700 that contains sample metadata boxes defined by the ISO Media File Format is extended to include sub-sample access boxes such as a sub-sample size box 702, a sub-sample description association box 704, a sub-sample to sample box 706 and a sub-sample description box 708. In one embodiment, the use of sub-sample access boxes is optional.

Referring to **Figure 7B**, a sample 710 may be, for example, divisible into slices such as a slice 712, data partitions such as partitions 714 and regions of interest (ROIs) such as a ROI 716. Each of these examples represents a different kind of division of samples into sub-samples. Sub-samples within a single sample may have different sizes.

A sub-sample size box 718 contains a version field that specifies the version of the sub-sample size box 718, a sub-sample size field specifying the default sub-sample size, a sub-sample count field to provide the number of sub-samples in the track, and an entry size field specifying the size of each sub-sample. If the sub-sample size field is set to 0, then the sub-samples have different sizes that are stored in the sub-sample size table 720. If the sub-sample size field is not set to 0, it specifies the constant sub-sample size, indicating that the sub-sample size table 720 is empty. The table 720 may have a fixed size of 32-bit or variable length field for representing the sub-sample sizes. If the field is varying length, the sub-sample table contains a field that indicates the length in bytes of the sub-sample size field.

Referring to **Figure 7C**, a sub-sample to sample box 722 includes a version field that specifies the version of the sub-sample to sample box 722, an entry count field that provides the number of entries in the table 723. Each entry in the sub-sample to sample table contains a first sample field that provides the index of the first sample in the run of

samples sharing the same number of sub-samples-per-sample, and a sub-samples per sample field that provides the number of sub-samples in each sample within a run of samples.

The table 723 can be used to find the total number of sub-samples in the track by computing how many samples are in a run, multiplying this number by the appropriate sub-samples-per-sample, and adding the results of all the runs together.

Referring to **Figure 7D**, a sub-sample description association box 724 includes a version field that specifies the version of the sub-sample description association box 724, a description type identifier that indicates the type of sub-samples being described (e.g., NAL packets, regions of interest, etc.), and an entry count field that provides the number of entries in the table 726. Each entry in table 726 includes a sub-sample description type identifier field indicating a sub-sample description ID and a first sub-sample field that gives the index of the first sub-sample in a run of sub-samples which share the same sub-sample description ID.

The sub-sample description type identifier controls the use of the sub-sample description ID field. That is, depending on the type specified in the description type identifier, the sub-sample description ID field may itself specify a description ID that directly encodes the sub-samples descriptions inside the ID itself or the sub-sample description ID field may serve as an index to a different table (i.e., a sub-sample description table described below)? For example, if the description type identifier indicates a JVT description, the sub-sample description ID field may include a code specifying the characteristics of JVT sub-samples. In this case, the sub-sample description ID field may be a 32-bit field, with the least significant 8 bits used as a bit-mask to represent the presence of predefined data partition inside a sub-sample and the higher order 24 bits used to represent the NAL packet type or for future extensions.

Referring to **Figure 7E**, a sub-sample description box 728 includes a version field that specifies the version of the sub-sample description box 728, an entry count field that provides the number of entries in the table 730, a description type identifier field that provides a description type of a sub-sample description field providing information about the characteristics of the sub-samples, and a table containing one or more sub-sample description entries 730. The sub-sample description type identifies the type to which the descriptive information relates and corresponds to the same field in the sub-sample description association table 724. Each entry in table 730 contains a sub-sample

description entry with information about the characteristics of the sub-samples associated with this description entry. The information and format of the description entry depend on the description type field. For example, when the description type is parameter set, then each description entry will contain the value of the parameter set.

The descriptive information may relate to parameter set information, information pertaining to ROI or any other information needed to characterize the sub-samples. For parameter sets, the sub-sample description association table 724 indicates the parameter set associated with each sub-sample. In such a case, the sub-sample description ID corresponds to the parameter set identifier. Similarly, a sub-sample can represent different regions-of-interest as follows. Define a sub-sample as one or more coded macroblocks and then use the sub-sample description association table to represent the division of the coded microblocks of a video frame or image into different regions. For example, the coded macroblocks in a frame can be divided into foreground and background macroblocks with two sub-sample description ID (e.g., sub-sample description IDs of 1 and 2), indicating assignment to the foreground and background regions, respectively.

Figure 7F illustrates different types of sub-samples. A sub-sample may represent a slice 732 with no partition, a slice 734 with multiple data partitions, a header 736 within a slice, a data partition 738 in the middle of a slice, the last data partition 740 of a slice, an SEI information packet 742, etc. Each of these sub-sample types may be associated with a specific value of an 8-bit mask 744 shown in **Figure 7G**. The 8-bit mask may form the 8 least significant bits of the 32-bit sub-sample description ID field as discussed above. **Figure 7H** illustrates the sub-sample description association box 724 having the description type identifier equal to "jvtd". The table 726 includes the 32-bit sub-sample description ID field storing the values illustrated in **Figure 7G**.

Figures 7H – 7K illustrate compression of data in a sub-sample description association table.

Referring to **Figure 7I**, an uncompressed table 726 includes a sequence 750 of sub-sample description IDs that repeats a sequence 748. In a compressed table 746, the repeated sequence 750 has been compressed into a reference to the sequence 748 and the number of times this sequence occurs.

In one embodiment illustrated in **Figure 7J**, a sequence occurrence can be encoded in the sub-sample description ID field by using its most significant bit as a run of sequence flag 754, its next 23 bits as an occurrence index 756, and its less significant bits as an

occurrence length 758. If the flag 754 is set to 1, then it indicates that this entry is an occurrence of a repeated sequence. Otherwise, this entry is a sub-sample description ID. The occurrence index 756 is the index in the sub-sample description association box 724 of the first occurrence of the sequence, and the length 758 indicates the length of the repeated sequence occurrence.

In another embodiment illustrated in **Figure 7K**, a repeated sequence occurrence table 760 is used to represent the repeated sequence occurrence. The most significant bit of the sub-sample description ID field is used as a run of sequence flag 762 indicating whether the entry is a sub-sample description ID or a sequence index 764 of the entry in the repeated sequence occurrence table 760 that is part of the sub-sample description association box 724. The repeated sequence occurrence table 760 includes an occurrence index field to specify the index in the sub-sample description association box 724 of the first item in the repeated sequence and a length field to specify the length of the repeated sequence.

Parameter Sets

In certain media formats, such as JVT, the "header" information containing the critical control values needed for proper decoding of media data are separated/decoupled from the rest of the coded data and stored in parameter sets. Then, rather than mixing these control values in the stream along with coded data, the coded data can refer to necessary parameter sets using a mechanism such as a unique identifier. This approach decouples the transmission of higher level coding parameters from coded data. At the same time, it also reduces redundancies by sharing common sets of control values as parameter sets.

To support efficient transmission of stored media streams that use parameter sets, a sender or player must be able to quickly link coded data to a corresponding parameter in order to know when and where the parameter set must be transmitted or accessed. One embodiment of the present invention provides this capability by storing data specifying the associations between parameter sets and corresponding portions of media data as parameter set metadata in a media file format.

Figures 8 and 9 illustrate processes for storing and retrieving parameter set metadata that are performed by the encoding system 100 and the decoding system 200

respectively. The processes may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both.

Figure 8 is a flow diagram of one embodiment of a method 800 for creating parameter set metadata at the encoding system 100. Initially, method 800 begins with processing logic receiving a file with encoded media data (processing block 802). The file includes sets of encoding parameters that specify how to decode portions of the media data. Next, processing logic examines the relationships between the sets of encoding parameters referred to as parameter sets and the corresponding portions of the media data (processing block 804) and creates parameter set metadata defining the parameter sets and their associations with the media data portions (processing block 806). The media data portions may be represented by samples or sub-samples.

In one embodiment, the parameter set metadata is organized into a set of predefined data structures (e.g., a set of boxes). The set of predefined data structures may include a data structure containing descriptive information about the parameter sets and a data structure containing information that defines associations between samples and corresponding parameter sets. In one embodiment, the set of predefined data structures also includes a data structure containing information that defines associations between sub-samples and corresponding parameter sets. The data structures containing sub-sample to parameter set association information may or may not override the data structures containing sample to parameter set association information.

Next, in one embodiment, processing logic determines whether any parameter set data structure contains a repeated sequence of data (decision box 808). If this determination is positive, processing logic converts each repeated sequence of data into a reference to a sequence occurrence and the number of times the sequence occurs (processing block 810).

Afterwards, at processing block 812, processing logic includes the parameter set metadata into a file associated with media data using a specific media file format (e.g., the JVT file format). Depending on the media file format, the parameter set metadata may be stored with track metadata and/or sample metadata (e.g., the data structure containing descriptive information about parameter sets may be included in a track box and the data structure(s) containing association information may be included in a sample table box) or independently from the track metadata and/or sample metadata.

Figure 9 is a flow diagram of one embodiment of a method 900 for utilizing parameter set metadata at the decoding system 200. Initially, method 900 begins with processing logic receiving a file associated with encoded media data (processing block 902). The file may be received from a database (local or external), the encoding system 100, or from any other device on a network. The file includes parameter set metadata that defines parameter sets for the media data and associations between the parameter sets and corresponding portions of the media data (e.g., corresponding samples or sub-samples).

Next, processing logic extracts the parameter set metadata from the file (processing block 904). As discussed above, the parameter set metadata may be stored in a set of data structures (e.g., a set of boxes).

Further, at processing block 906, processing logic uses the extracted metadata to determine which parameter set is associated with a specific media data portion (e.g., a sample or a sub-sample). This information may then be used to control transmission time of media data portions and corresponding parameter sets. That is, a parameter set that is to be used to decode a specific sample or sub-sample must be sent prior to a packet containing the sample or sub-sample or with the packet containing the sample or sub-sample.

Accordingly, the use of parameter set metadata enables independent transmission of parameter sets on a more reliable channel, reducing the chance of errors or data loss causing parts of the media stream to be lost.

Exemplary parameter set metadata structures will now be described with reference to an extended ISO media file format (referred to as an extended ISO). It should be noted, however, that other media file formats can be extended to incorporate various data structures for storing parameter set metadata.

Figures 10A – 10E illustrate exemplary data structures for storing parameter set metadata.

Referring to **Figure 10A**, a track box 1002 that contains track metadata boxes defined by the ISO file format is extended to include a parameter set description box 1004. In addition, a sample table box 1006 that contains sample metadata boxes defined by ISO file format is extended to include a sample to parameter set box 1008. In one embodiment, the sample table box 1006 includes a sub-sample to parameter set box which may override the sample to parameter set box 1008 as will be discussed in more detail below.

In one embodiment, the parameter set metadata boxes 1004 and 1008 are mandatory. In another embodiment, only the parameter set description box 1004 is mandatory. In yet another embodiment, all of the parameter set metadata boxes are optional.

Referring to **Figure 10B**, a parameter set description box 1010 contains a version field that specifies the version of the parameter set description box 1010, a parameter set description count field to provide the number of entries in a table 1012, and a parameter set entry field containing entries for the parameter sets themselves.

Parameter sets may be referenced from the sample level or the sub-sample level. Referring to **Figure 10C**, a sample to parameter set box 1014 provides references to parameter sets from the sample level. The sample to parameter set box 1014 includes a version field that specifies the version of the sample to parameter set box 1014, a default parameter set ID field that specifies the default parameter set ID, an entry count field that provides the number of entries in the table 1016. Each entry in table 1016 contains a first sample field providing the index of a first sample in a run of samples that share the same parameter set, and a parameter set index specifying the index to the parameter set description box 1010. If the default parameter set ID is equal to 0, then the samples have different parameter sets that are stored in the table 1016. Otherwise, a constant parameter set is used and no array follows.

In one embodiment, data in the table 1016 is compressed by converting each repeated sequence into a reference to an initial sequence and the number of times this sequence occurs, as discussed in more detail above in conjunction with the sub-sample description association table.

Parameter sets may be referenced from the sub-sample level by defining associations between parameter sets and sub-samples. In one embodiment, the associations between parameter sets and sub-samples are defined using a sub-sample description association box described above. **Figure 10D** illustrates a sub-sample description association box 1018 with the description type identifier referring to parameter sets (e.g., the description type identifier is equal to "pars"). Based on this description type identifier, the sub-sample description ID in the table 1020 indicates the index in the parameter set description box 1010.

In one embodiment, when the sub-sample description association box 1018 with the description type identifier referring to parameter sets is present, it overrides the sample to parameter set box 1014.

A parameter set may change between the time the parameter set is created and the time the parameter set is used to decode a corresponding portion of media data. If such a change occurs, the decoding system 200 receives a parameter update packet specifying a change to the parameter set. The parameter set metadata includes data identifying the state of the parameter set both before the update and after the update.

Referring to **Figure 10E**, the parameter set description box 1010 includes an entry for the initial parameter set 1022 created at time t_0 and an entry for an updated parameter set 1024 created in response to a parameter update packet 1026 received at time t_1. The sub-sample description association box 1018 associates the two parameter sets with corresponding sub-samples.

Sample Groups

While the samples within a track can have various logical groupings (partitions) of samples into sequences (possibly non-consecutive) that represent high-level structures in the media data, existing file formats do not provide convenient mechanisms for representing and storing such groupings. For example, advanced coding formats such as JVT organize samples within a single track into groups based on their inter-dependencies. These groups (referred to herein as sequences or sample groups) may be used to identify chains of disposable samples when required by network conditions, thus supporting temporal scalability. Storing metadata that defines sample groups in a file format enables the sender of the media to easily and efficiently implement the above features.

An example of a sample group is a set of samples whose inter-frame dependencies allow them to be decoded independently of other samples. In JVT, such a sample group is referred to as an enhanced group of pictures (enhanced GOP). In an enhanced GOP, samples may be divided into sub-sequences. Each sub-sequence includes a set of samples that depend on each other and can be disposed of as a unit. In addition, samples of an enhanced GOP may be hierarchically structured into layers such that samples in a higher layer are predicted only from samples in a lower layer, thus allowing the samples of the

highest layer to be disposed of without affecting the ability to decode other samples. The lowest layer that includes samples that do not depend on samples in any other layers is referred to as a base layer. Any other layer that is not the base layer is referred to as an enhancement layer.

Figure 11 illustrates an exemplary enhanced GOP in which the samples are divided into two layers, a base layer 1102 and an enhancement layer 1104, and two sub-sequences 1106 and 1108. Each of the two sub-sequences 1106 and 1108 can be dropped independently of each other.

Figures 12 and 13 illustrate processes for storing and retrieving sample group metadata that are performed by the encoding system 100 and the decoding system 200 respectively. The processes may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both.

Figure 12 is a flow diagram of one embodiment of a method 1200 for creating sample group metadata at the encoding system 100. Initially, method 1200 begins with processing logic receiving a file with encoded media data (processing block 1202). Samples within a track of the media data have certain inter-dependencies. For example, the track may include I-frames that do not depend on any other samples, P-frames that depend on a single prior sample, and B-frames that depend on two prior samples including any combination of I-frames, P-frames and B-frames. Based on their inter-dependencies, samples in a track can be logically combined into sample groups (e.g., enhanced GOPs, layers, sub-sequences, etc.).

Next, processing logic examines the media data to identify sample groups in each track (processing block 1204) and creates sample group metadata that describes the sample groups and defines which samples are contained in each sample group (processing block 1206). In one embodiment, the sample group metadata is organized into a set of predefined data structures (e.g., a set of boxes). The set of predefined data structures may include a data structure containing descriptive information about each sample group and a data structure containing information that identifies samples contained in each sample group.

Next, in one embodiment, processing logic determines whether any sample group data structure contains a repeated sequence of data (decision box 1208). If this determination is positive, processing logic converts each repeated sequence of data into a

reference to a sequence occurrence and the number of times the sequence occurs (processing block 1210).

Afterwards, at processing block 1212, processing logic includes the sample group metadata into a file associated with media data using a specific media file format (e.g., the JVT file format). Depending on the media file format, the sample group metadata may be stored with sample metadata (e.g., the sample group data structures may be included in a sample table box) or independently from the sample metadata.

Figure 13 is a flow diagram of one embodiment of a method 1300 for utilizing sample group metadata at the decoding system 200. Initially, method 1300 begins with processing logic receiving a file associated with encoded media data (processing block 1302). The file may be received from a database (local or external), the encoding system 100, or from any other device on a network. The file includes sample group metadata that defines sample groups in the media data.

Next, processing logic extracts the sample group metadata from the file (processing block 1304). As discussed above, the sample group metadata may be stored in a set of data structures (e.g., a set of boxes).

Further, at processing block 1306, processing logic uses the extracted sample group metadata to identify chains of samples that can be disposed of without affecting the ability to decode other samples. In one embodiment, this information may be used to access samples in a specific sample group and determine which samples can be dropped in response to a change in network capacity. In other embodiments, sample group metadata is used to filter samples so that only a portion of the samples in a track are processed or rendered.

Accordingly, the sample group metadata facilitates selective access to samples and scalability.

Exemplary sample group metadata structures will now be described with reference to an extended ISO media file format (referred to as an extended MP4). It should be noted, however, that other media file formats can be extended to incorporate various data structures for storing sample group metadata.

Figures 14A – 14E illustrate exemplary data structures for storing sample group metadata.

Referring to **Figure 14A**, a sample table box 1400 that contains sample metadata boxes defined by MP4 is extended to include a sample group box 1402 and a sample

group description box 1404. In one embodiment, the sample group metadata boxes 1402 and 1404 are optional.

Referring to **Figure 14B**, a sample group box 1406 is used to find a set of samples contained in a particular sample group. Multiple instances of the sample group box 1406 are allowed to correspond to different types of sample groups (e.g., enhanced GOPs, sub-sequences, layers, parameter sets, etc.). The sample group box 1406 contains a version field that specifies the version of the sample group box 1406, an entry count field to provide the number of entries in a table 1408, a sample group identifier field to identify the type of the sample group, a first sample field providing the index of a first sample in a run of samples that are contained in the same sample group, and a sample group description index specifying the index to a sample group description box.

Referring to **Figure 14C**, a sample group description box 1410 provides information about the characteristics of a sample group. The sample group description box 1410 contains a version field that specifies the version of the sample group description box 1410, an entry count field to provide the number of entries in a table 1412, a sample group identifier field to identify the type of the sample group, and a sample group description field to provide sample group descriptors.

Referring to **Figure 14D**, the use of the sample group box 1416 for the layers ("layr") sample group type is illustrated. Samples 1 through 11 are divided into three layers based on the samples' inter-dependencies. In layer 0 (the base layer), samples (samples 1, 6 and 11) depend only on each other but not on samples in any other layers. In layer 1, samples (samples 2, 5, 7, 10) depend on samples in the lower layer (i.e., layer 0) and samples within this layer 1. In layer 2, samples (samples 3, 4, 8, 9) depend on samples in lower layers (layers 0 and 1) and samples within this layer 2. Accordingly, the samples of layer 2 can be disposed of without affecting the ability to decode samples from lower layers 0 and 1.

Data in the sample group box 1416 illustrates the above associations between the samples and the layers. As shown, this data includes a repetitive layer pattern 1414 which can be compressed by converting each repeated layer pattern into a reference to an initial layer pattern and the number of times this pattern occurs, as discussed in more detail above.

Referring to **Figure 14E**, the use of a sample group box 1418 for the sub-sequence ("sseq") sample group type is illustrated. Samples 1 through 11 are divided into four sub-

sequences based on the samples' inter-dependencies. Each sub-sequence, except sub-sequence 0 at layer 0, includes samples on which no other sub-sequences depend. Thus, the samples in the sub-sequence can be disposed of as a unit when needed.

Data in the sample group box 1418 illustrates associations between the samples and the sub-sequences. This data allows random access to samples at the beginning of a corresponding sub-sequence.

Stream Switching

In typical streaming scenarios, one of the key requirements is to scale the bit rate of the compressed data in response to changing network conditions. The simplest way to achieve this is to encode multiple streams with different bit-rates and quality settings for representative network conditions. The server can then switch amongst these pre-coded streams in response to network conditions.

The JVT standard provides a new type of picture, called switching pictures that allow one picture to reconstruct identically to another without requiring the two pictures to use the same frame for prediction. In particular, JVT provides two types of switching pictures: SI-pictures, which, like I-frames, are coded independent of any other pictures; and SP-pictures, which are coded with reference to other pictures. Switching pictures can be used to implement switching amongst streams with different bit-rates and quality setting in response to changing delivery conditions, to provide error resilience, and to implement trick modes like fast forward and rewind.

However, to use JVT switching pictures effectively when implementing stream switching, error resilience, trick modes, and other features, the player has to know which samples in the stored media data have the alternate representations and what their dependencies are. Existing file formats do not provide such capability.

One embodiment of the present invention addresses the above limitation by defining switch sample sets. A switch sample set represents a set of samples whose decoded values are identical but which may use different reference samples. A reference sample is a sample used to predict the value of another sample. Each member of a switch sample set is referred to as a switch sample. **Figure 15A** illustrate the use of a switch sample set for bit stream switching.

Referring to **Figure 15A**, stream 1 and stream 2 are two encodings of the same content with different quality and bit-rate parameters. Sample S12 is a SP-picture, not

occurring in either stream, that is used to implement switching from stream 1 to stream 2 (switching is a directional property). Samples S12 and S2 are contained in a switch sample set. Both S1 and S12 are predicted from sample P12 in track 1 and S2 is predicted from sample P22 in track 2. Although samples S12 and S2 use different reference samples, their decoded values are identical. Accordingly, switching from stream 1 to stream 2 (at sample S1 in stream 1 and S2 in stream 2) can be achieved via switch sample S12.

Figures 16 and 17 illustrate processes for storing and retrieving switch sample metadata that are performed by the encoding system 100 and the decoding system 200 respectively. The processes may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both.

Figure 16 is a flow diagram of one embodiment of a method 1600 for creating switch sample metadata at the encoding system 100. Initially, method 1600 begins with processing logic receiving a file with encoded media data (processing block 1602). The file includes one or more alternate encodings for the media data (e.g., for different bandwidth and quality settings for representative network conditions). The alternate encodings includes one or more switching pictures. Such pictures may be included inside the alternate media data streams or as separate entities that implement special features such as error resilience or trick modes. The method for creating these tracks and switch pictures is not specified by this invention but various possibilities would be obvious to one versed in the art. For example, the periodic (e.g., every 1 second) placement of switch samples between each pair of tracks containing alternate encodings.

Next, processing logic examines the file to create switch sample sets that include those samples having the same decoding values while using different reference samples (processing block 1604) and creates switch sample metadata that defines switch sample sets for the media data and describes samples within the switch sample sets (processing block 1606). In one embodiment, the switch sample metadata is organized into a predefined data structure such as a table box containing a set of nested tables.

Next, in one embodiment, processing logic determines whether the switch sample metadata structure contains a repeated sequence of data (decision box 1608). If this determination is positive, processing logic converts each repeated sequence of data into a

reference to a sequence occurrence and the number of times the sequence occurs (processing block 1610).

Afterwards, at processing block 1612, processing logic includes the switch sample metadata into a file associated with media data using a specific media file format (e.g., the JVT file format). In one embodiment, the switch sample metadata may be stored in a separate track designated for stream switching. In another embodiment, the switch sample metadata is stored with sample metadata (e.g., the sequences data structures may be included in a sample table box).

Figure 17 is a flow diagram of one embodiment of a method 1700 for utilizing switch sample metadata at the decoding system 200. Initially, method 1700 begins with processing logic receiving a file associated with encoded media data (processing block 1702). The file may be received from a database (local or external), the encoding system 100, or from any other device on a network. The file includes switch sample metadata that defines switch sample sets associated with the media data.

Next, processing logic extracts the switch sample metadata from the file (processing block 1704). As discussed above, the switch sample metadata may be stored in a data structure such as a table box containing a set of nested tables.

Further, at processing block 1706, processing logic uses the extracted metadata to find a switch sample set that contains a specific sample and select an alternative sample from the switch sample set. The alternative sample, which has the same decoding value as the initial sample, may then be used to switch between two differently encoded bit streams in response to changing network conditions, to provide random access entry point into a bit stream, to facilitate error recovery, etc.

An exemplary switch sample metadata structure will now be described with reference to an extended ISO media file format (referred to as an extended MP4). It should be noted, however, that other media file formats could be extended to incorporate various data structures for storing switch sample metadata.

Figure 18 illustrates an exemplary data structure for storing switch sample metadata. The exemplary data structure is in the form of a switch sample table box that includes a set of nested tables. Each entry in a table 1802 identifies one switch sample set. Each switch sample set consists of a group of switch samples whose reconstruction is objectively identical (or perceptually identical) but which may be predicted from different reference samples that may or may not be in the same track (stream) as the switch sample.

Each entry in the table 1802 is linked to a corresponding table 1804. The table 1804 identifies each switch sample contained in a switch sample set. Each entry in the table 1804 is further linked to a corresponding table 1806 which defines the location of a switch sample (i.e., its track and sample number), the track containing reference samples used by the switch sample, the total number of reference samples used by the switch sample, and each reference sample used by the switch sample.

As illustrated in **Figure 15A**, in one embodiment, the switch sample metadata may be used to switch between differently encoded versions of the same content. In MP4, each alternate coding is stored as a separate MP4 track and the "alternate group" in the track header indicates that it is an alternate encoding of specific content.

Figure 15B illustrates a table containing metadata that defines a switch sample set 1502 consisting of samples S2 and S12 according to **Figure 15A**.

Figure 15C is a flow diagram of one embodiment of a method 1510 for determining a point at which a switch between two bit streams is to be performed. Assuming that the switch is to be performed from stream 1 to stream 2, method 1510 begins with searching switch sample metadata to find all switch sample sets that contain a switch sample with a reference track of stream 1 and a switch sample with a switch sample track of stream 2 (processing block 1512). Next, the resulting switch sample sets are evaluated to select a switch sample set in which all reference samples of a switch sample with the reference track of stream 1 are available (processing block 1514). For example, if the switch sample with the reference track of stream 1 is a P frame, one sample before switching is required to be available. Further, the samples in the selected switch sample set are used to determine the switching point (processing block 1516). That is, the switching point is considered to be immediately after the highest reference sample of the switch sample with the reference track of stream 1, via the switch sample with the reference track of stream 1, and to the sample immediately following the switch sample with the switch sample track of stream 2.

In another embodiment, switch sample metadata may be used to facilitate random access entry points into a bit stream as illustrated in **Figures 19A – 19C**.

Referring to **Figures 19A and 19B**, a switch sample 1902 consists of samples S2 and S12. S2 is a P-frame predicted from P22 and used during usual stream playback. S12 is used as a random access point (e.g., for splicing). Once S12 is decoded, stream playback continues with decoding of P24 as if P24 was decoded after S2.

Figure 19C is a flow diagram of one embodiment of a method 1910 for determining a random access point for a sample (e.g., sample S on track T). Method 1910 begins with searching switch sample metadata to find all switch sample sets that contain a switch sample with a switch sample track T (processing block 1912). Next, the resulting switch sample sets are evaluated to select a switch sample set in which a switch sample with the switch sample track T is the closest sample prior to sample S in decoding order (processing block 1914). Further, a switch sample (sample SS) other than the switch sample with the switch sample track T is chosen from the selected switch sample set for a random access point to sample S (processing block 1916). During stream playback, sample SS is decoded (following by the decoding of any reference samples specified in the entry for sample SS) instead of sample S.

In yet another embodiment, switch sample metadata may be used to facilitate error recovery as illustrated in **Figures 20A – 20C**.

Referring to **Figures 20A and 20B**, a switch sample 2002 consists of samples S2, S12 and S22. Sample S2 is predicted from sample P4. Sample S12 is predicted from sample S1. If an error occurs between samples P2 and P4, the switch sample S12 can be decoded instead of sample S2. Streaming then continues with sample P6 as usual. If an error affects sample S1 as well, switch sample S22 can be decoded instead of sample S2, and then streaming will continue with sample P6 as usual.

Figure 20C is a flow diagram of one embodiment of a method 2010 for facilitating error recovery when sending a sample (e.g., sample S). Method 2010 begins with searching switch sample metadata to find all switch sample sets that contain a switch sample equal to sample S or following sample S in the decoding order (processing block 2012). Next, the resulting switch sample sets are evaluated to select a switch sample set with a switch sample SS that is the closest to sample S and whose reference samples are known (via feedback or some other information source) to be correct (processing block 2014). Further, switch sample SS is sent instead of sample S (processing block 2016).

Storage and retrieval of audiovisual metadata has been described. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention.

CLAIMS

We claim:

1. A method comprising:

creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data; and

forming a file associated with the multimedia data, the file comprising the parameter set metadata.

2. The method of claim 1 wherein each of the plurality of portions of multimedia data is a sample within the multimedia data.

3. The method of claim 1 wherein each of the plurality of portions of multimedia data is a sub-sample within a portion of the multimedia data.

4. The method of claim 1 wherein creating parameter set metadata comprises:

receiving a file with encoded multimedia data;

examining relationships between the one or more parameter sets and the plurality of portions of multimedia data; and

defining the parameter set metadata based on the examined relationships.

5. The method of claim 1 wherein creating parameter set metadata comprises:

organizing the parameter set metadata into a set of predefined data structures.

6. The method of claim 5 wherein creating parameter set metadata further comprises:

converting each repeated sequence of data within the set of predefined data structures into a reference to a sequence occurrence and a number of occurrences.

7. The method of claim 5 wherein the set of predefined data structures comprises a first data structure containing descriptive information about the one or more parameter sets

and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of multimedia data.

8. The method of claim 1 further comprising:

sending the file associated with the multimedia data to a decoding system;

receiving the file associated with the multimedia data at the decoding system; and

extracting, at the decoding system, the parameter set metadata from the file associated with the multimedia data, the extracted parameter set metadata being subsequently used to identify any of the one or more parameter sets that are required to decode at least a portion of the multimedia data.

9. A method comprising:

receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data; and

extracting the parameter set metadata from the file, the extracted parameter set metadata being subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data.

10. The method of claim 9 wherein each of the plurality of portions of the multimedia data is a sample within the multimedia data.

11. The method of claim 9 wherein each of the plurality of portions of the multimedia data is a sub-sample within a portion of the multimedia data.

12. The method of claim 9 further comprising:

controlling transmission time for the plurality of portions of the multimedia data and the one or more parameter sets using the determined relationships.

13. The method of claim 9 wherein the extracted parameter set metadata is organized into a set of predefined data structures.

14. The method of claim 13 wherein the set of predefined data structures comprises a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of the multimedia data.

15. A method comprising:

creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data;

creating sample group metadata defining groupings of a plurality of samples within the multimedia data; and

forming a file associated with the multimedia data, the file comprising the parameter set metadata and the sample group metadata.

16. The method of claim 15 wherein each of the plurality of portions of multimedia data is any one of a sample and sub-sample within the multimedia data.

17. The method of claim 15 wherein creating parameter set metadata comprises:

organizing the parameter set metadata into a set of predefined data structures comprising a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of multimedia data..

18. The method of claim 15 wherein the groupings are based on inter-dependencies of the plurality of samples.

19. The method of claim 15 wherein creating sample group metadata comprises:

organizing the sample group metadata into a set of predefined data structures, comprising a first data structure containing descriptive information about a plurality of sample groups within the multimedia data and a second data structure containing information that identifies samples in each of the plurality of sample groups.

20. A method comprising:

receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data and sample group metadata defining groupings of a plurality of samples within the multimedia data; and

extracting the parameter set metadata and the sample group metadata from the file, the extracted parameter set metadata being subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data and the extracted sample group metadata being subsequently used to identify samples that can be disposed of in future processing.

21. The method of claim 20 wherein each of the plurality of portions of the multimedia data is any one of a sample and a sub-sample within the multimedia data.

22. The method of claim 20 further comprising:

controlling transmission time for the plurality of portions of the multimedia data and the one or more parameter sets using the determined relationships.

23. The method of claim 20 wherein the extracted parameter set metadata is organized into a set of predefined data structures comprising a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of the multimedia data.

24. The method of claim 20 wherein the groupings are based on inter-dependencies of the plurality of samples.

25. The method of claim 20 further comprising:

finding, in response to a change in network capacity, one or more samples that can be disposed of without affecting decoding of the remaining samples of the multimedia data.

26. The method of claim 20 further comprising:

filtering, based on the extracted sample group metadata, the plurality of samples to reduce a number of samples that will be rendered.

27. The method of claim 20 wherein the extracted sample group metadata is organized into a set of predefined data structures comprising a first data structure containing descriptive information about a plurality of sample groups within the multimedia data and a second data structure containing information that identifies samples in each of the plurality of sample groups.

28. A method comprising:

creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data;

creating sample group metadata defining groupings of a plurality of samples within the multimedia data;

creating switch sample metadata defining a plurality of switch sample sets associated with the multimedia data, each of the plurality of switch sample sets containing samples that have identical decoding values; and

forming a file associated with the multimedia data, the file comprising the parameter set metadata, the sample group metadata and the switch sample metadata.

29. The method of claim 28 wherein each of the plurality of portions of multimedia data is any one of a sample and sub-sample within the multimedia data.

30. The method of claim 28 wherein creating parameter set metadata comprises:

organizing the parameter set metadata into a set of predefined data structures comprising a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of multimedia data..

31. The method of claim 28 wherein the groupings are based on inter-dependencies of the plurality of samples.

32. The method of claim 28 wherein creating sample group metadata comprises:

organizing the sample group metadata into a set of predefined data structures, comprising a first data structure containing descriptive information about a plurality of sample groups within the multimedia data and a second data structure containing information that identifies samples in each of the plurality of sample groups.

33. The method of claim 28 wherein the samples in each of the plurality of switch sample sets use different reference samples.

34. The method of claim 28 wherein creating switch sample metadata comprises:

organizing the switch sample metadata into a predefined data structure represented as a table box containing a set of nested tables.

35. A method comprising:

receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data, sample group metadata defining groupings of a plurality of samples within the multimedia data, and switch sample metadata defining a plurality of switch sample sets associated with the multimedia data; and

extracting the parameter set metadata, the sample group metadata and the switch sample metadata from the file, the extracted parameter set metadata being subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data, the extracted sample group metadata being subsequently used to identify samples that can be disposed of in future processing, and the extracted switch sample metadata being subsequently used to find a replacement for a specific sample.

36. The method of claim 35 wherein each of the plurality of portions of the multimedia data is any one of a sample and a sub-sample within the multimedia data.

37. The method of claim 35 further comprising:

controlling transmission time for the plurality of portions of the multimedia data and the one or more parameter sets using the determined relationships.

38. The method of claim 35 wherein the extracted parameter set metadata is organized into a set of predefined data structures comprising a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of the multimedia data.

39. The method of claim 35 wherein the groupings are based on inter-dependencies of the plurality of samples.

40. The method of claim 35 further comprising:

finding, in response to a change in network capacity, one or more samples that can be disposed of without affecting decoding of the remaining samples of the multimedia data.

41. The method of claim 35 further comprising:

filtering, based on the extracted sample group metadata, the plurality of samples to reduce a number of samples that will be rendered.

42. The method of claim 35 wherein the extracted sample group metadata is organized into a set of predefined data structures comprising a first data structure containing descriptive information about a plurality of sample groups within the multimedia data and a second data structure containing information that identifies samples in each of the plurality of sample groups.

43. The method of claim 35 wherein each of the plurality of switch sample sets contains samples that have the same decoding value while using different reference samples.

44. The method of claim 35 further comprising:

finding, in the plurality of switch sample sets, a switch sample set that contains a specific sample; and

selecting an alternative sample from the found switch sample set.

45. The method of claim 35 wherein the extracted switch sample metadata is organized into a predefined data structure represented as a table box containing a set of nested tables.

46. A method comprising:

creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data;

creating switch sample metadata defining a plurality of switch sample sets associated with the multimedia data, each of the plurality of switch sample sets containing samples that have identical decoding values; and

forming a file associated with the multimedia data, the file comprising the parameter set metadata and the switch sample metadata.

47. The method of claim 46 wherein each of the plurality of portions of multimedia data is any one of a sample and a sub-sample within the multimedia data.

48. The method of claim 46 wherein creating parameter set metadata comprises:

organizing the parameter set metadata into a set of predefined data structures comprising a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of multimedia data..

49. The method of claim 46 wherein the samples in each of the plurality of switch sample sets use different reference samples.

50. The method of claim 46 wherein creating switch sample metadata comprises:

organizing the switch sample metadata into a predefined data structure represented as a table box containing a set of nested tables.

51. A method comprising:

receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data and switch sample metadata defining a plurality of switch sample sets associated with the multimedia data; and

extracting the parameter set metadata and the switch sample metadata from the file, the extracted parameter set metadata being subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data, and the extracted switch sample metadata being subsequently used to find a replacement for a specific sample.

52. The method of claim 51 wherein each of the plurality of portions of the multimedia data is any one of a sample and a sub-sample within the multimedia data.

53. The method of claim 51 further comprising:

controlling transmission time for the plurality of portions of the multimedia data and the one or more parameter sets using the determined relationships.

54. The method of claim 51 wherein the extracted parameter set metadata is organized into a set of predefined data structures comprising a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of the multimedia data.

55. The method of claim 51 wherein each of the plurality of switch sample sets contains samples that have the same decoding value while using different reference samples.

56. The method of claim 51 further comprising:

finding, in the plurality of switch sample sets, a switch sample set that contains a specific sample; and

selecting an alternative sample from the found switch sample set.

57. The method of claim 51 wherein the extracted switch sample metadata is organized into a predefined data structure represented as a table box containing a set of nested tables.

58. A memory for storing data for access by an application program being executed on a data processing system, comprising:

a plurality of data structures stored in said memory, said plurality of data structures being resident in a file used by said application program, said file being associated with multimedia data and including parameter set metadata defining one or more parameter sets for a plurality of portions of the multimedia data.

59. The memory of claim 58 wherein the file including the parameter set metadata also includes the associated multimedia data.

60. The memory of claim 58 wherein the file including the parameter set metadata contains references to a file containing the associated multimedia data.

61. The memory of claim 58 wherein the plurality of data structures comprises a first data structure containing descriptive information about the one or more parameter sets and a second data structure containing information that defines associations between the one or more parameter sets and the plurality of portions of the multimedia data.

62. A memory for storing data for access by an application program being executed on a data processing system, comprising:

a plurality of data structures stored in said memory, said plurality of data structures being resident in a file used by said application program, said file being associated with multimedia data and including

parameter set metadata defining one or more parameter sets for a plurality of portions of the multimedia data, and

sample group metadata defining groupings of a plurality of samples within the multimedia data.

63. A memory for storing data for access by an application program being executed on a data processing system, comprising:

a plurality of data structures stored in said memory, said plurality of data structures being resident in a file used by said application program, said file being associated with multimedia data and including

parameter set metadata defining one or more parameter sets for a plurality of portions of the multimedia data,

sample group metadata defining groupings of a plurality of samples within the multimedia data, and

switch sample metadata defining a plurality of switch sample sets associated with the multimedia data.

64. A memory for storing data for access by an application program being executed on a data processing system, comprising:

a plurality of data structures stored in said memory, said plurality of data structures being resident in a file used by said application program, said file being associated with multimedia data and including

parameter set metadata defining one or more parameter sets for a plurality of portions of the multimedia data, and

switch sample metadata defining a plurality of switch sample sets associated with the multimedia data.

65. An apparatus comprising:

a metadata generator to create parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data; and

a file creator to form a file associated with the multimedia data, the file comprising the parameter set metadata.

66. The apparatus of claim 65 wherein each of the plurality of portions of multimedia data is any one of a sample and a sub-sample within the multimedia data.

67. The apparatus of claim 65 wherein the metadata generator is to create parameter set metadata by receiving a file with encoded multimedia data, examining relationships between the one or more parameter sets and the plurality of portions of multimedia data, and defining the parameter set metadata based on the examined relationships.

68. The apparatus of claim 65 further comprising:

a metadata extractor to receive the file associated with the multimedia data at the decoding system and to extract the parameter set metadata from the file associated with the multimedia data; and

a media data stream processor to use the extracted parameter set metadata for identifying any of the one or more parameter sets that are required to decode at least a portion of the multimedia data.

69. An apparatus comprising:

metadata extractor to receive a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data, and to extract the parameter set metadata from the file; and

a media data stream processor to use the extracted parameter set metadata for determining relationships between the one or more parameter sets and a plurality of portions of the multimedia data.

70. The apparatus of claim 69 wherein each of the plurality of portions of the multimedia data is a sample or a sub-sample within the multimedia data.

71. The apparatus of claim 69 wherein the media data stream processor is further to control transmission time for the plurality of portions of the multimedia data and the one or more parameter sets using the determined relationships.

72. An apparatus comprising:

a metadata generator to create parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data and to create sample group metadata defining groupings of a plurality of samples within the multimedia data; and

a file creator to form a file associated with the multimedia data, the file comprising the parameter set metadata and the sample group metadata.

73. An apparatus comprising:

a metadata extractor to receive a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data and sample group metadata defining groupings of a plurality of samples within the multimedia data, and to extract the parameter set metadata and the sample group metadata from the file; and

a media data stream processor to use the extracted parameter set metadata for determining relationships between the one or more parameter sets and a plurality of portions of the multimedia data and to use the extracted sample group metadata for identifying samples that can be disposed of in future processing.

74. An apparatus comprising:

a metadata generator to create parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data, to create sample group metadata defining groupings of a plurality of samples within the multimedia data, and to create switch sample metadata defining a plurality of switch sample sets associated with the multimedia data; and

a file creator to form a file associated with the multimedia data, the file comprising the parameter set metadata, the sample group metadata and the switch sample metadata.

75. An apparatus comprising:

a metadata extractor to receive a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data, sample group metadata defining groupings of a plurality of samples within the multimedia data, and switch sample metadata defining a plurality of switch sample sets associated with the multimedia data, and to extract the parameter set metadata, the sample group metadata and the switch sample metadata from the file; and

a media data stream processor to use the extracted parameter set metadata for determining relationships between the one or more parameter sets and a plurality of portions of the multimedia data, to use the extracted sample group metadata for identifying samples that can be disposed of in future processing, and to use the extracted switch sample metadata for finding a replacement for a specific sample.

76. An apparatus comprising:

a metadata generator to create parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data and to create switch sample metadata defining a plurality of switch sample sets associated with the multimedia data; and

a file creator to form a file associated with the multimedia data, the file comprising the parameter set metadata and the switch sample metadata.

77. An apparatus comprising:

a metadata extractor to receive a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data and switch sample metadata defining a plurality of switch sample sets associated with the multimedia data, and to extract the parameter set metadata and the switch sample metadata from the file; and

a media data stream processor to use the extracted parameter set metadata for determining relationships between the one or more parameter sets and a plurality of portions of the multimedia data, and to use the extracted switch sample metadata for finding a replacement for a specific sample.

78. An apparatus comprising:

means for creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data; and

means for forming a file associated with the multimedia data, the file comprising the parameter set metadata.

79. An apparatus comprising:

means for receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data; and

means for extracting the parameter set metadata from the file, the extracted parameter set metadata being subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data.

80. An apparatus comprising:

means for creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data;

means for creating sample group metadata defining groupings of a plurality of samples within the multimedia data; and

means for forming a file associated with the multimedia data, the file comprising the parameter set metadata and the sample group metadata.

81. An apparatus comprising:

means for receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data and sample group metadata defining groupings of a plurality of samples within the multimedia data; and

means for extracting the parameter set metadata and the sample group metadata from the file, the extracted parameter set metadata being subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data and the extracted sample group metadata being subsequently used to identify samples that can be disposed of in future processing.

82. An apparatus comprising:

means for creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data;

means for creating sample group metadata defining groupings of a plurality of samples within the multimedia data;

means for creating switch sample metadata defining a plurality of switch sample sets associated with the multimedia data, each of the plurality of switch sample sets containing samples that have identical decoding values; and

means for forming a file associated with the multimedia data, the file comprising the parameter set metadata, the sample group metadata and the switch sample metadata.

83. An apparatus comprising:

means for receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data, sample group metadata defining groupings of a plurality of samples within the multimedia data, and switch sample metadata defining a plurality of switch sample sets associated with the multimedia data; and

means for extracting the parameter set metadata, the sample group metadata and the switch sample metadata from the file, the extracted parameter set metadata being

subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data, the extracted sample group metadata being subsequently used to identify samples that can be disposed of in future processing, and the extracted switch sample metadata being subsequently used to find a replacement for a specific sample.

84. An apparatus comprising:

means for creating parameter set metadata identifying one or more parameter sets for a plurality of portions of multimedia data;

means for creating switch sample metadata defining a plurality of switch sample sets associated with the multimedia data, each of the plurality of switch sample sets containing samples that have identical decoding values; and

means for forming a file associated with the multimedia data, the file comprising the parameter set metadata and the switch sample metadata.

85. An apparatus comprising:

means for receiving a file associated with multimedia data, the file comprising parameter set metadata identifying one or more parameter sets for the multimedia data and switch sample metadata defining a plurality of switch sample sets associated with the multimedia data; and

means for extracting the parameter set metadata and the switch sample metadata from the file, the extracted parameter set metadata being subsequently used to determine relationships between the one or more parameter sets and a plurality of portions of the multimedia data, and the extracted switch sample metadata being subsequently used to find a replacement for a specific sample.

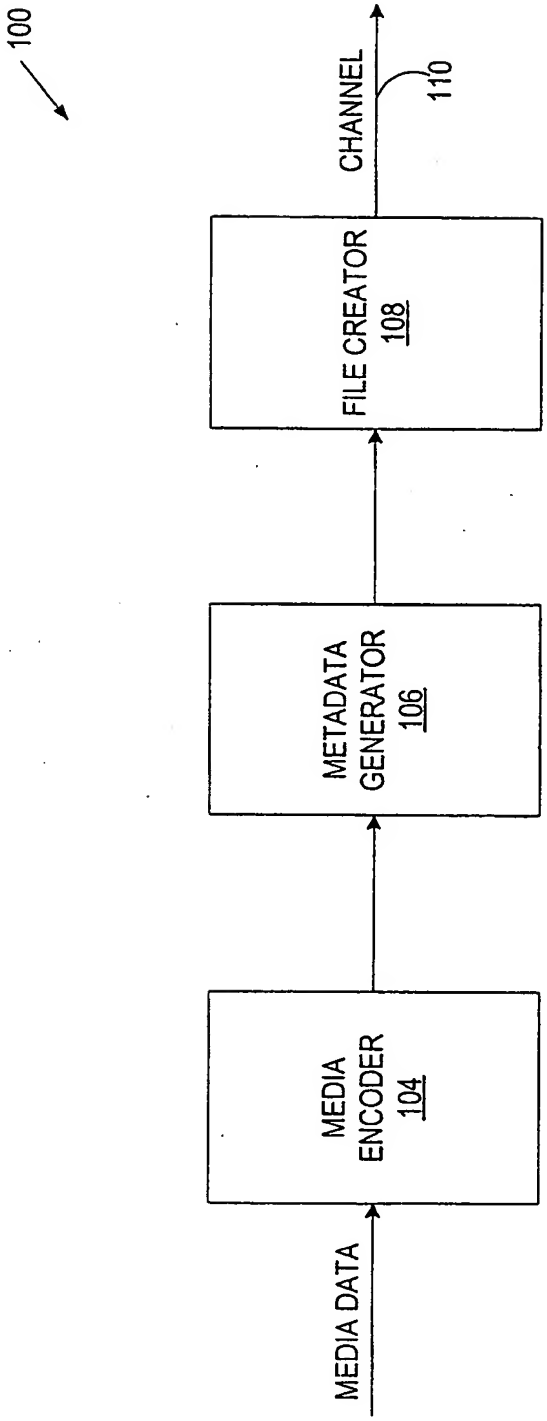

100
CHANNEL
110
FILE CREATOR
108
METADATA
GENERATOR
106
MEDIA
ENCODER
104
MEDIA DATA

FIG. 1

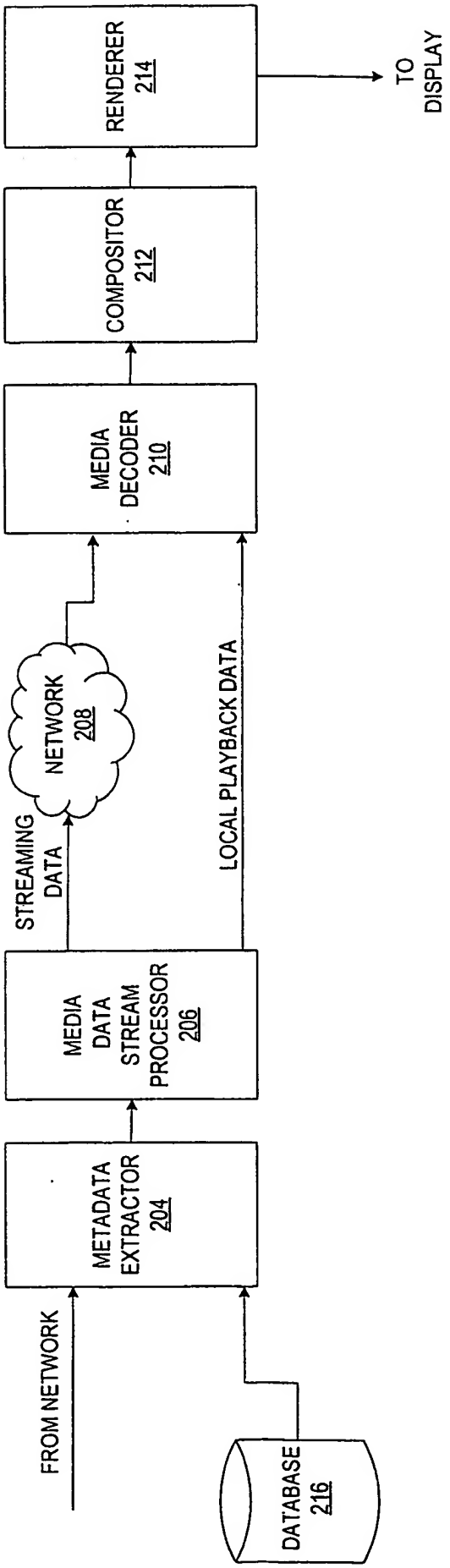

RENDERER 214
TO DISPLAY
COMPOSITOR 212
MEDIA DECODER 210
NETWORK 208
STREAMING DATA
LOCAL PLAYBACK DATA
MEDIA DATA STREAM PROCESSOR 206
METADATA EXTRACTOR 204
FROM NETWORK
DATABASE 216

FIG. 2

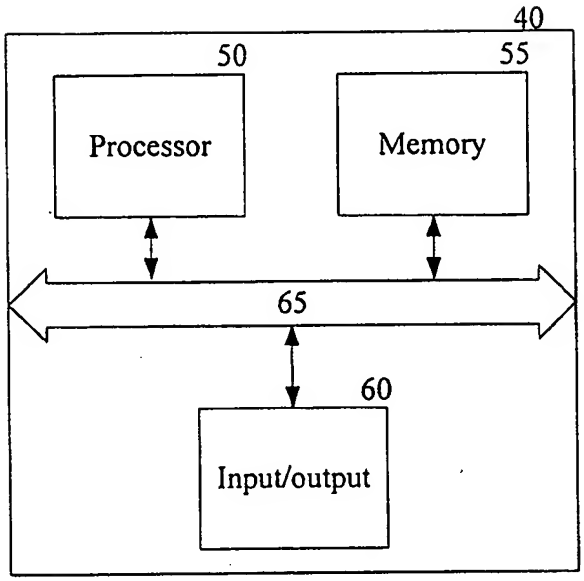


FIGURE 3

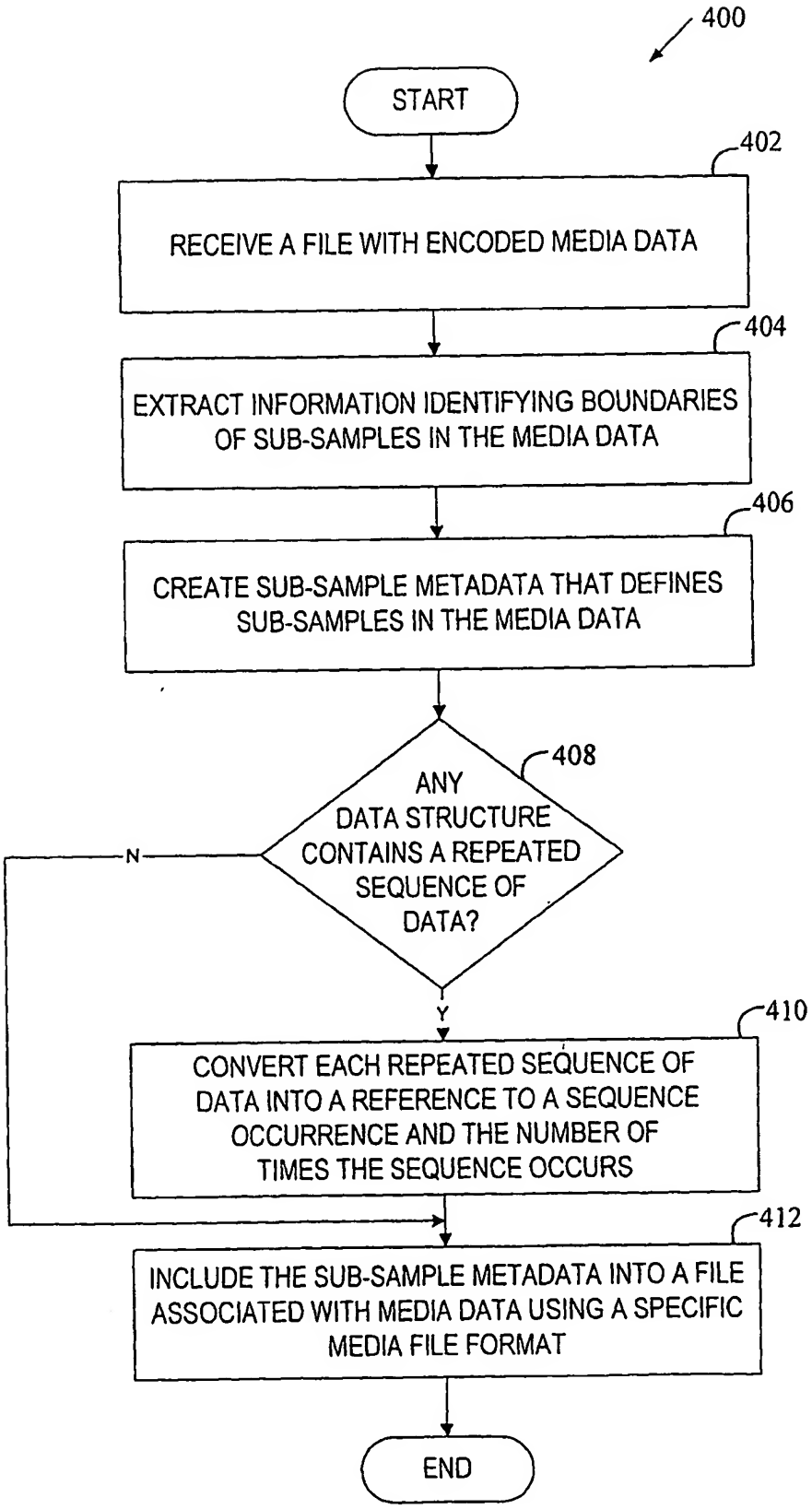

400
START
402
RECEIVE A FILE WITH ENCODED MEDIA DATA
404
EXTRACT INFORMATION IDENTIFYING BOUNDARIES OF SUB-SAMPLES IN THE MEDIA DATA
406
CREATE SUB-SAMPLE METADATA THAT DEFINES SUB-SAMPLES IN THE MEDIA DATA
408
ANY DATA STRUCTURE CONTAINS A REPEATED SEQUENCE OF DATA?
N
Y
410
CONVERT EACH REPEATED SEQUENCE OF DATA INTO A REFERENCE TO A SEQUENCE OCCURRENCE AND THE NUMBER OF TIMES THE SEQUENCE OCCURS
412
INCLUDE THE SUB-SAMPLE METADATA INTO A FILE ASSOCIATED WITH MEDIA DATA USING A SPECIFIC MEDIA FILE FORMAT
END

FIG. 4

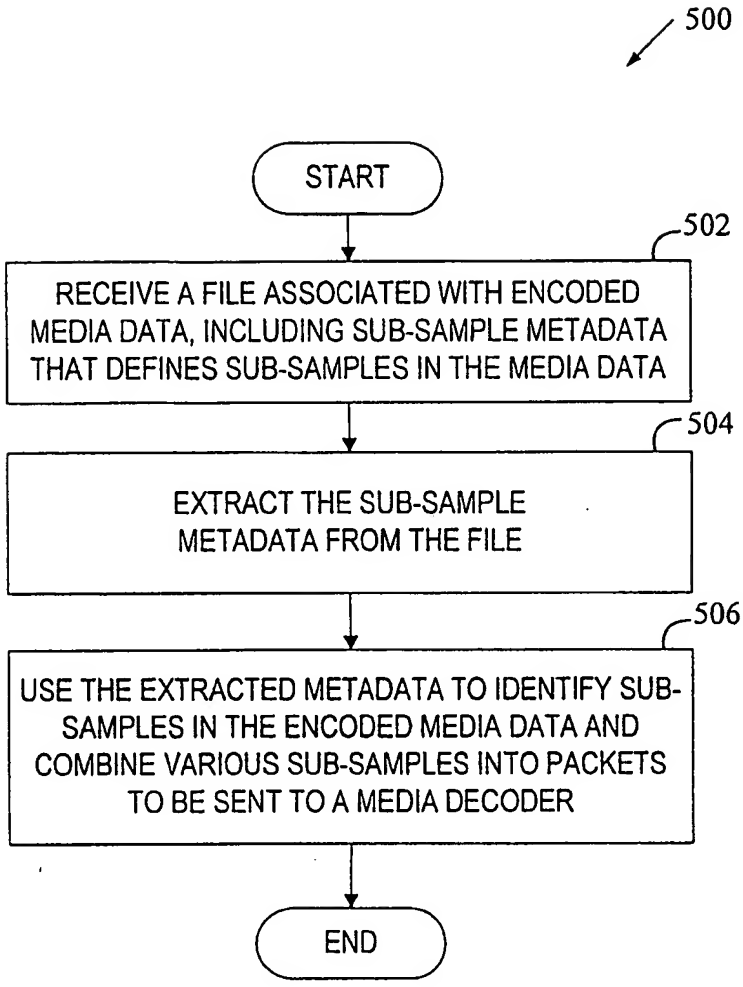

500
START
502
RECEIVE A FILE ASSOCIATED WITH ENCODED MEDIA DATA, INCLUDING SUB-SAMPLE METADATA THAT DEFINES SUB-SAMPLES IN THE MEDIA DATA
504
EXTRACT THE SUB-SAMPLE METADATA FROM THE FILE
506
USE THE EXTRACTED METADATA TO IDENTIFY SUB-SAMPLES IN THE ENCODED MEDIA DATA AND COMBINE VARIOUS SUB-SAMPLES INTO PACKETS TO BE SENT TO A MEDIA DECODER
END

FIG. 5

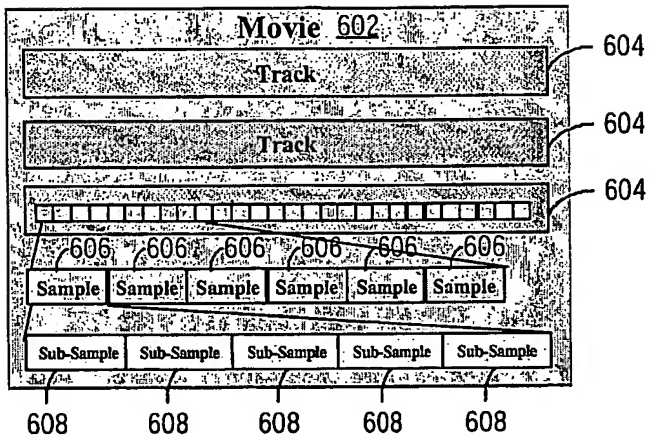

Movie 602
Track
Track
604
604
604
606
606
606
606
606
606
Sample
Sample
Sample
Sample
Sample
Sample
Sub-Sample
Sub-Sample
Sub-Sample
Sub-Sample
Sub-Sample
608
608
608
608
608

FIG. 6

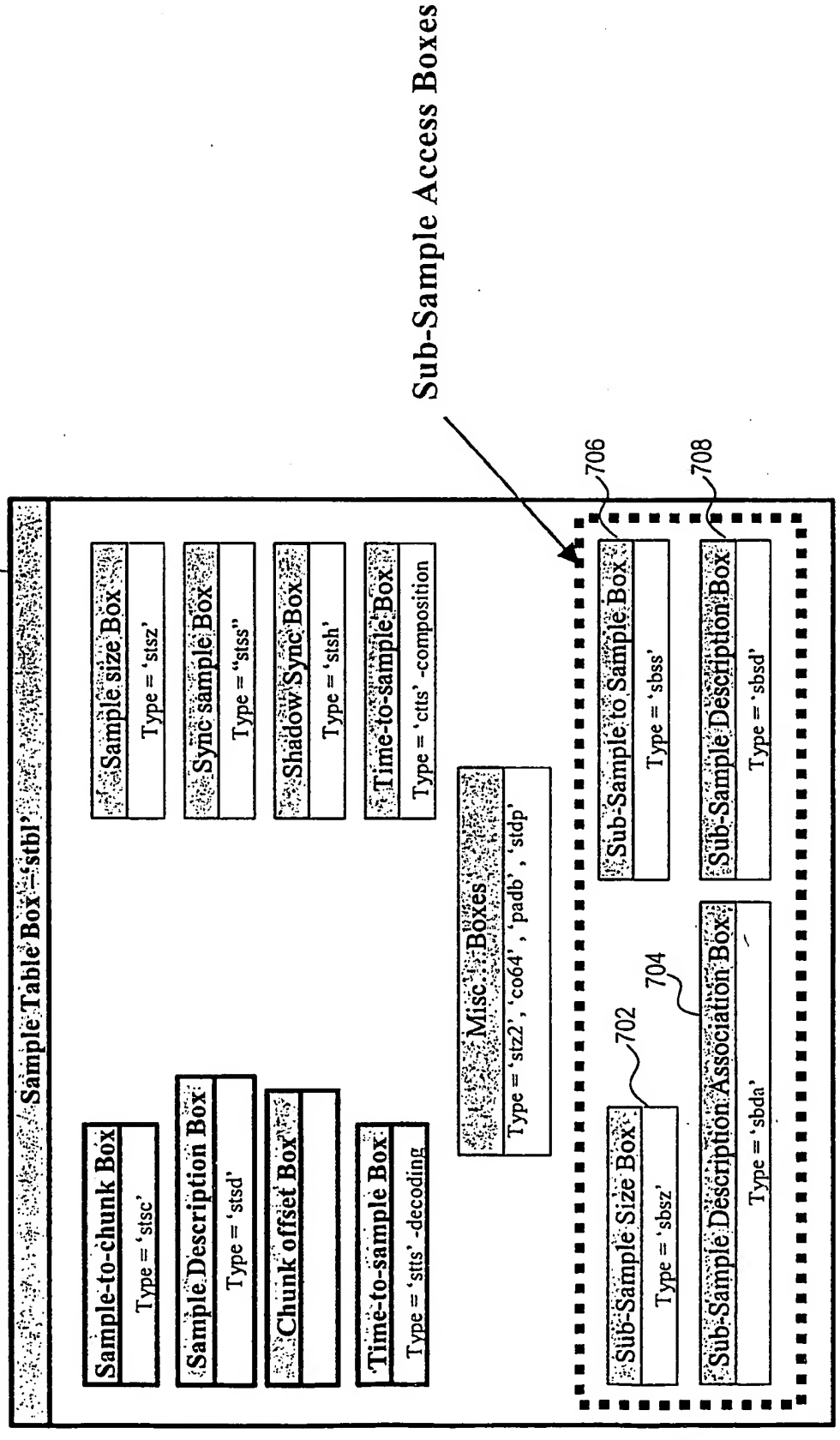

Sub-Sample Access Boxes
700
Sample Table Box – 'stbl'
Sample-to-chunk Box
Type = 'stsc'
Sample Description Box
Type = 'stsd'
Chunk offset Box
Time-to-sample Box
Type = 'stts' -decoding
Sample size Box
Type = 'stsz'
Sync sample Box
Type = "stss"
Shadow Sync Box
Type = 'stsh'
Time-to-sample Box
Type = 'ctts' -composition
Misc. Boxes
Type = 'stz2', 'co64', 'padb', 'stdp'
706
Sub-Sample to Sample Box
Type = 'sbss'
708
Sub-Sample Description Box
Type = 'sbsd'
Sub-Sample Size Box
Type = 'sbsz'
702
704
Sub-Sample Description Association Box
Type = 'sbda'

FIG. 7A

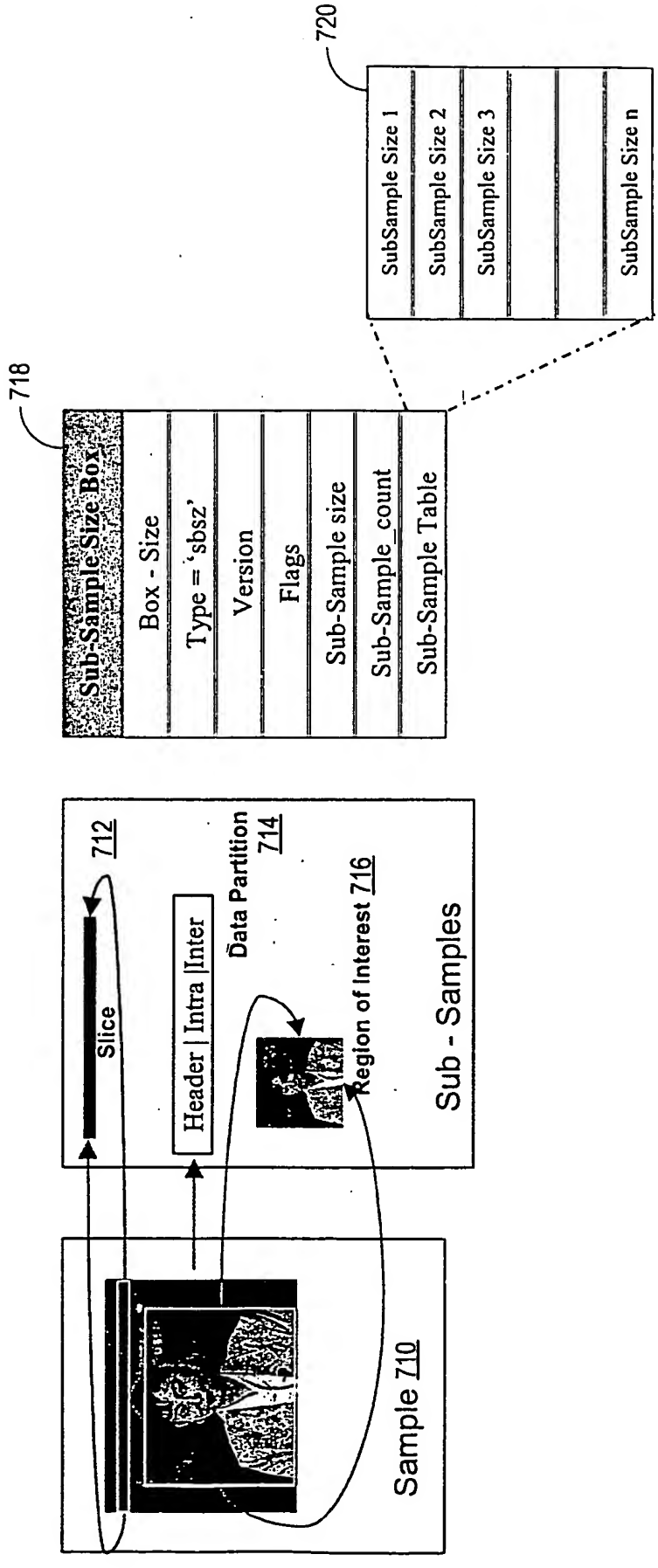

720
SubSample Size 1
SubSample Size 2
SubSample Size 3
SubSample Size n
718
Sub-Sample Size Box
Box - Size
Type = 'sbsz'
Version
Flags
Sub-Sample size
Sub-Sample_count
Sub-Sample Table
712
Slice
Header | Intra | Inter
Data Partition 714
Region of Interest 716
Sub - Samples
Sample 710

FIG. 7B

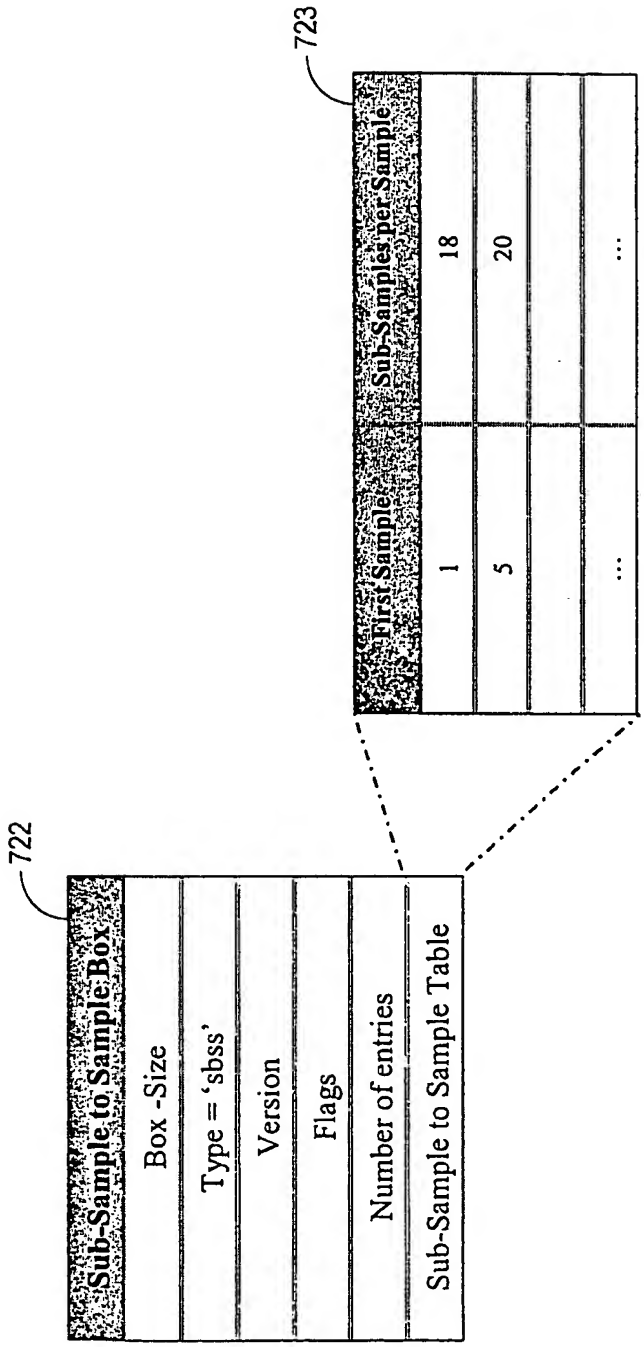

722
Sub-Sample to Sample Box
Box -Size
Type = 'sbss'
Version
Flags
Number of entries
Sub-Sample to Sample Table
723
First Sample
Sub-Samples per Sample
1
18
5
20
...
...

FIG. 7C

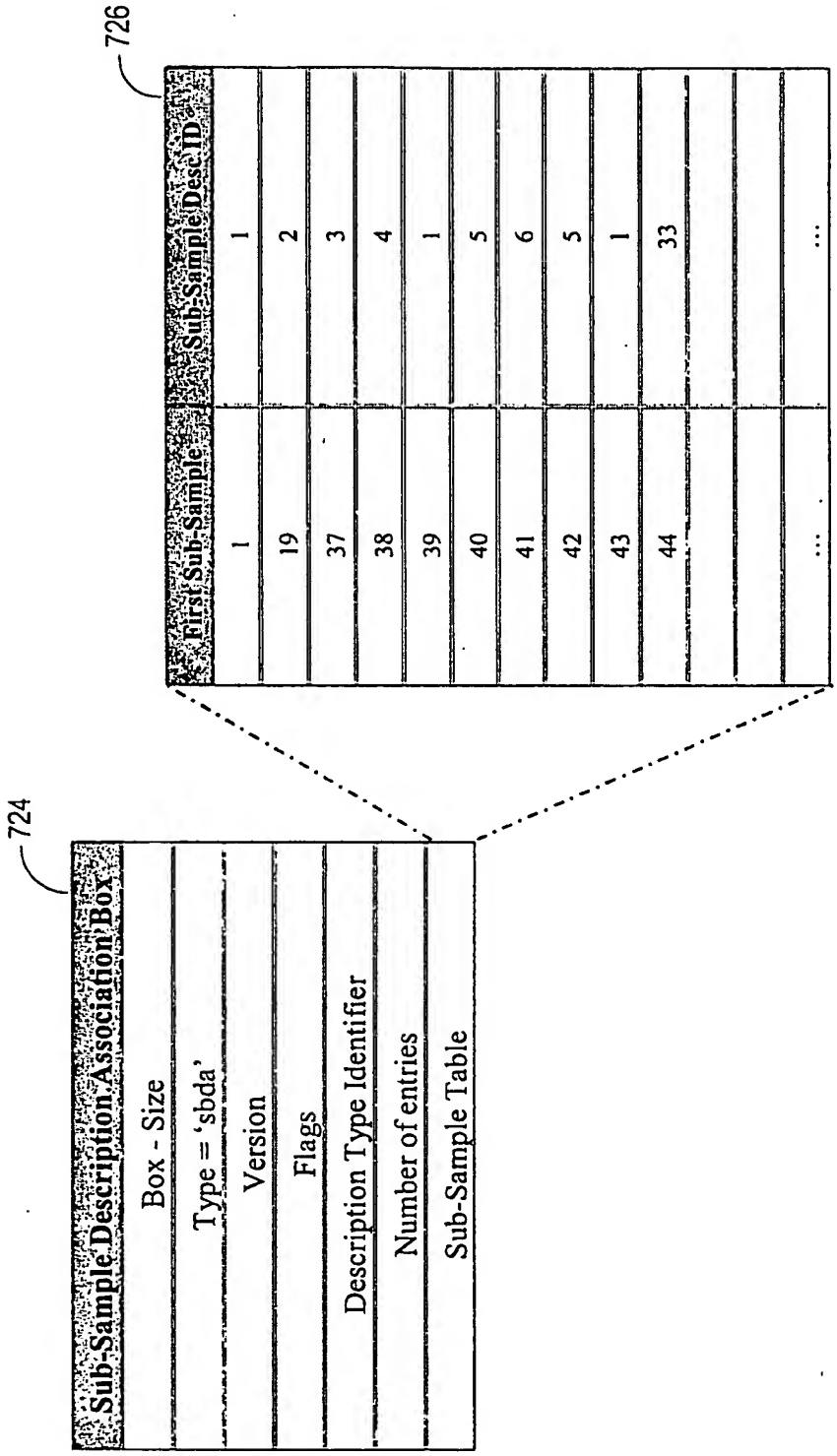

724
Sub-Sample Description Association Box
Box - Size
Type = 'sbda'
Version
Flags
Description Type Identifier
Number of entries
Sub-Sample Table
726
First Sub-Sample
Sub-Sample Desc ID
1 1
19 2
37 3
38 4
39 1
40 5
41 6
42 5
43 1
44 33
... ...

FIG. 7D

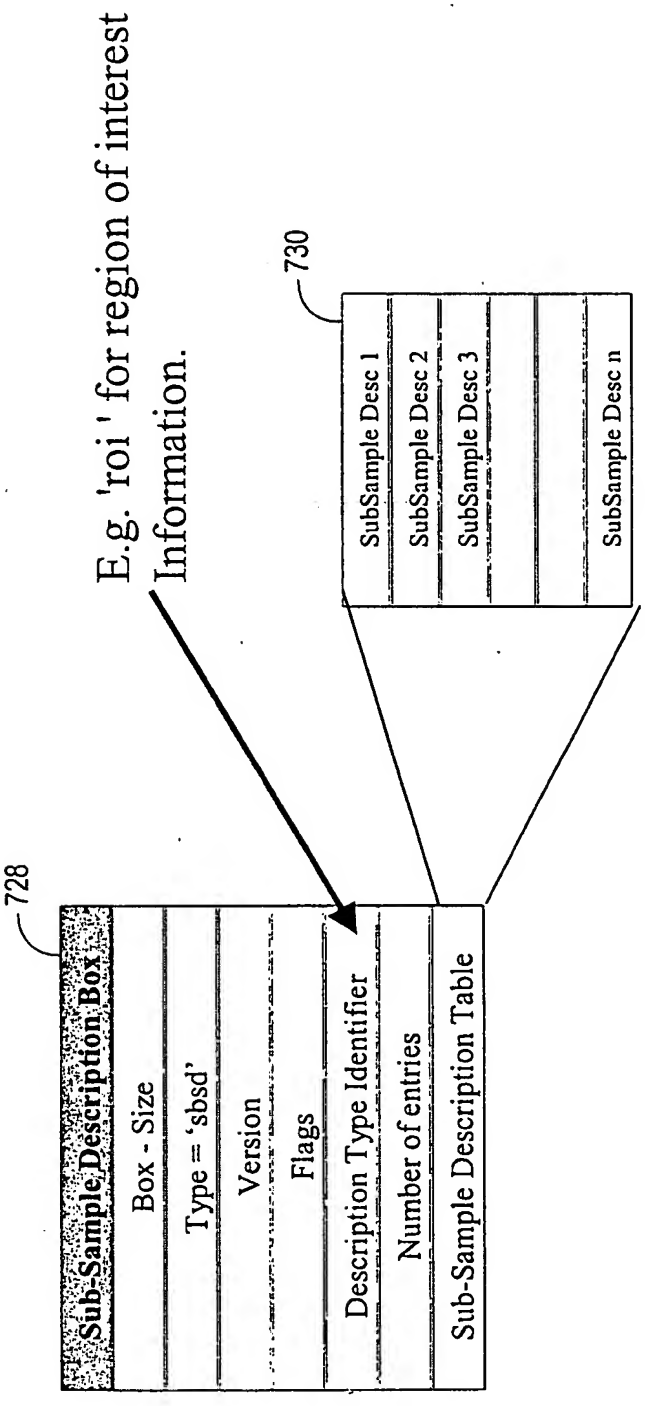

728
Sub-Sample Description Box
Box - Size
Type = 'sbsd'
Version
Flags
Description Type Identifier
Number of entries
Sub-Sample Description Table
730
SubSample Desc 1
SubSample Desc 2
SubSample Desc 3
SubSample Desc n
E.g. 'roi ' for region of interest Information.

FIG. 7E

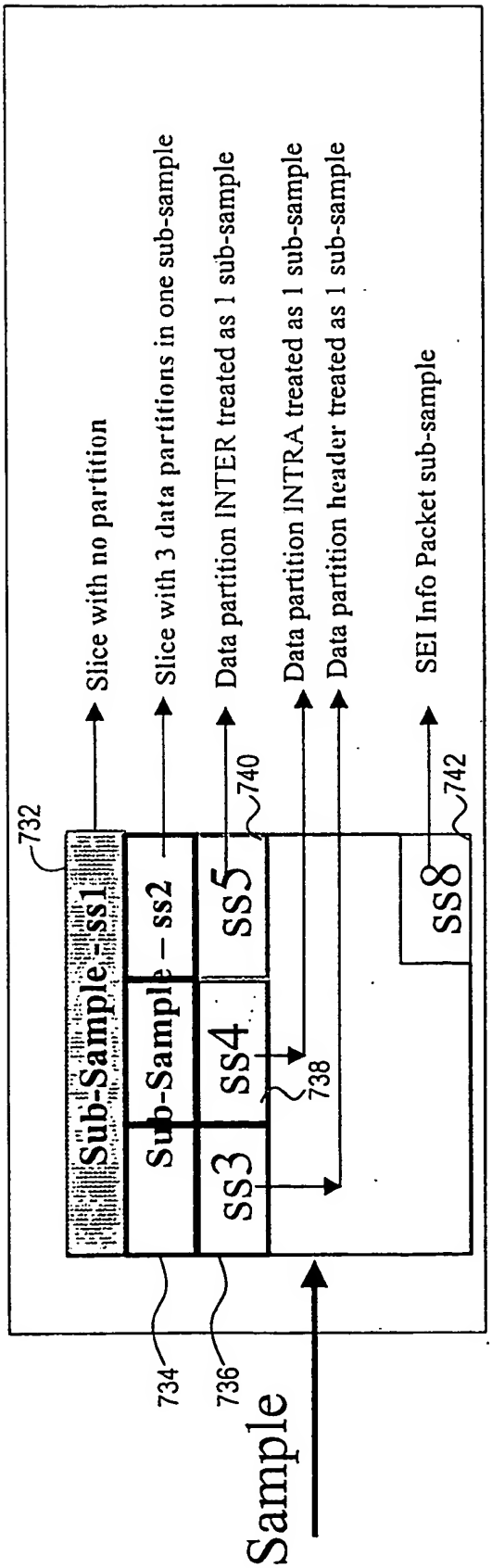

Sample
734
736
732
Sub-Sample - ss1
Sub-Sample – ss2
ss3
ss4
ss5
ss8
738
740
742
Slice with no partition
Slice with 3 data partitions in one sub-sample
Data partition INTER treated as 1 sub-sample
Data partition INTRA treated as 1 sub-sample
Data partition header treated as 1 sub-sample
SEI Info Packet sub-sample

FIG. 7F

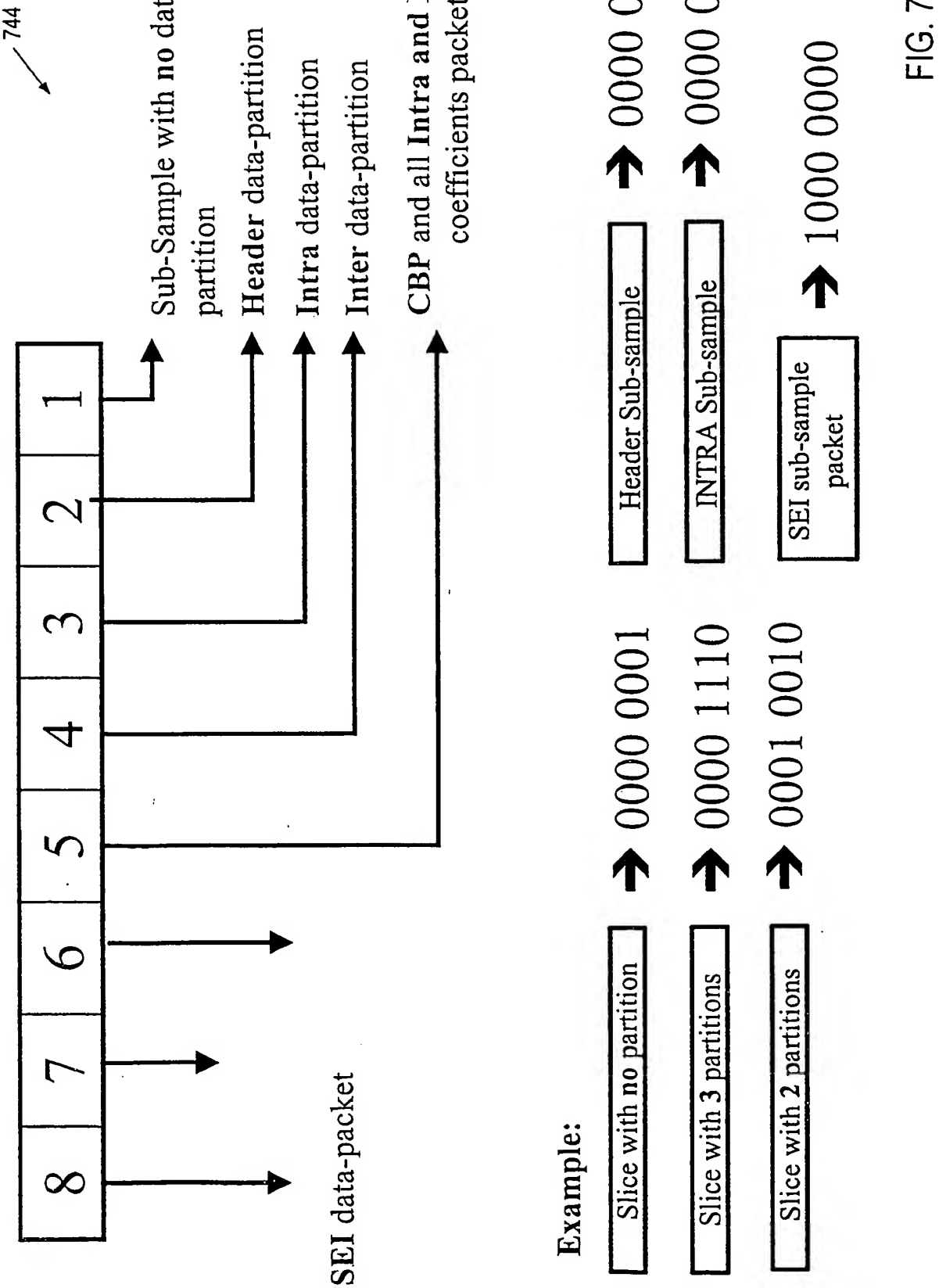

744
8
7
6
5
4
3
2
1
SEI data-packet
CBP and all Intra and Inter coefficients packet
Inter data-partition
Intra data-partition
Header data-partition
Sub-Sample with no data-partition
Example:
Slice with no partition
0000 0001
Header Sub-sample
0000 0010
Slice with 3 partitions
0000 1110
INTRA Sub-sample
0000 0100
Slice with 2 partitions
0001 0010
SEI sub-sample packet
1000 0000

FIG. 7G

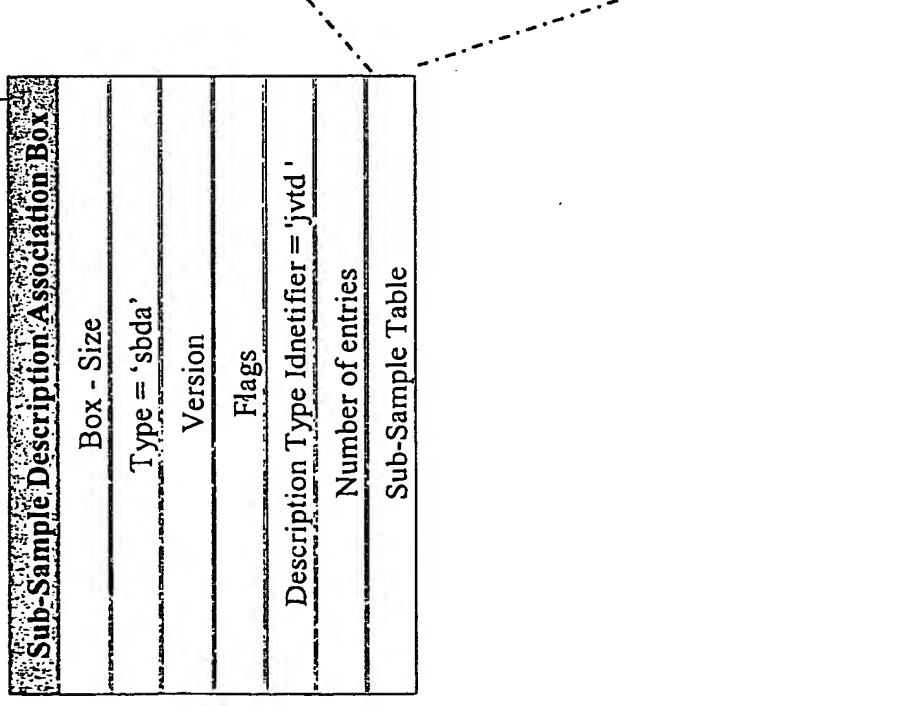

724
Sub-Sample Description Association Box
Box - Size
Type = 'sbda'
Version
Flags
Description Type Idnetifier = 'jvtd'
Number of entries
Sub-Sample Table
726
32 bits
First Sub-Sample
Sub-Sample Desc ID
1 ..... 0000 0001
19 ..... 0000 1110
37 ..... 0001 0010
38 ..... 1000 0000
39 ..... 0000 0010
40 ..... 0000 0100
41 ..... 0000 1000
42 ..... 0000 0010
43 ..... 0000 0100
44 ..... 0000 1000
...
no data -partition
data –partitioned slices
SEI packet
Header packet
INTRA packet
INTER packet
Header packet
INTRA packet
INTER packet

FIG. 7H

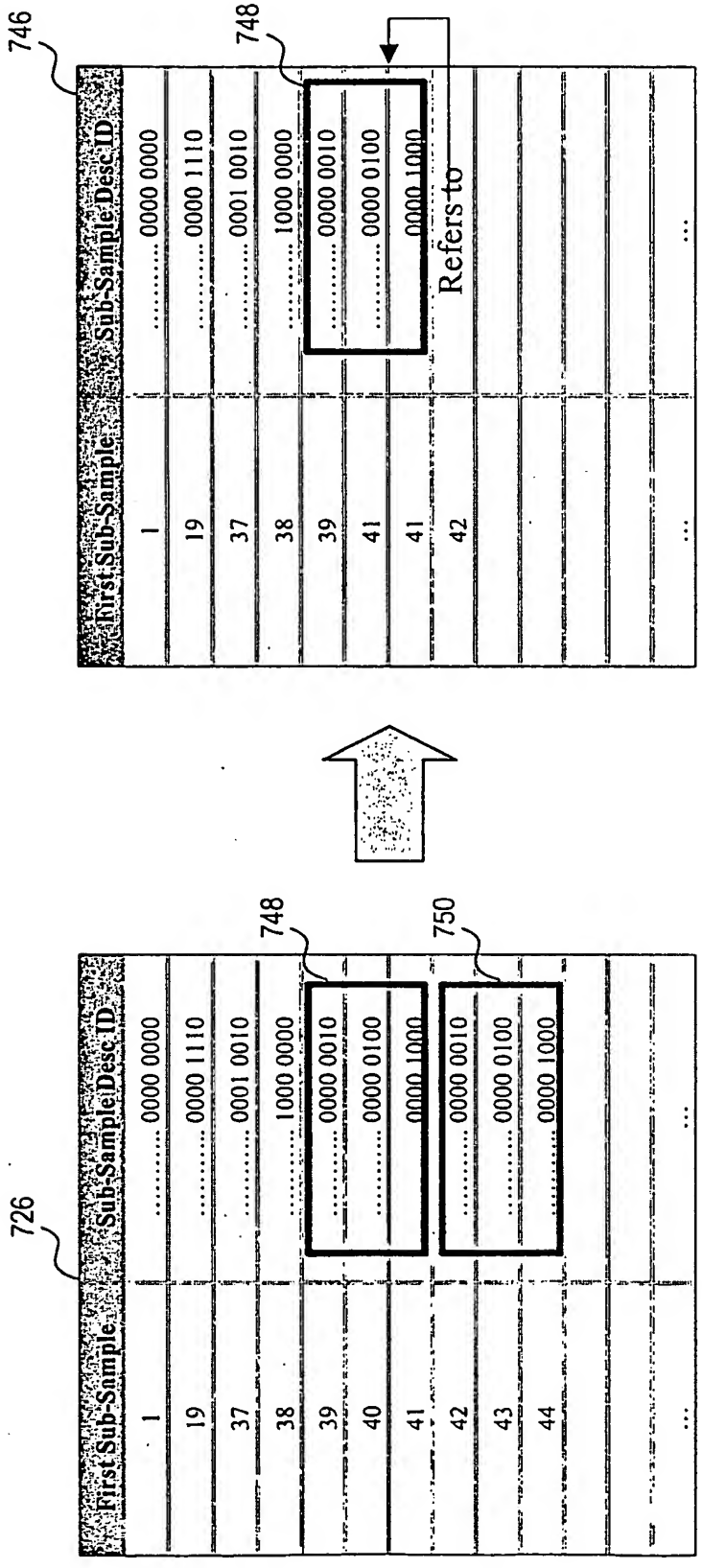

726
First Sub-Sample
Sub-Sample Desc ID
1 ..... 0000 0000
19 ..... 0000 1110
37 ..... 0001 0010
38 ..... 1000 0000
39 ..... 0000 0010
40 ..... 0000 0100
41 ..... 0000 1000
42 ..... 0000 0010
43 ..... 0000 0100
44 ..... 0000 1000
748
750
Uncompressed Table
746
First Sub-Sample
Sub-Sample Desc ID
1 ..... 0000 0000
19 ..... 0000 1110
37 ..... 0001 0010
38 ..... 1000 0000
39 ..... 0000 0010
41 ..... 0000 0100
41 ..... 0000 1000
42
748
Refers to
Compressed with sequence repetition.

FIG. 7I

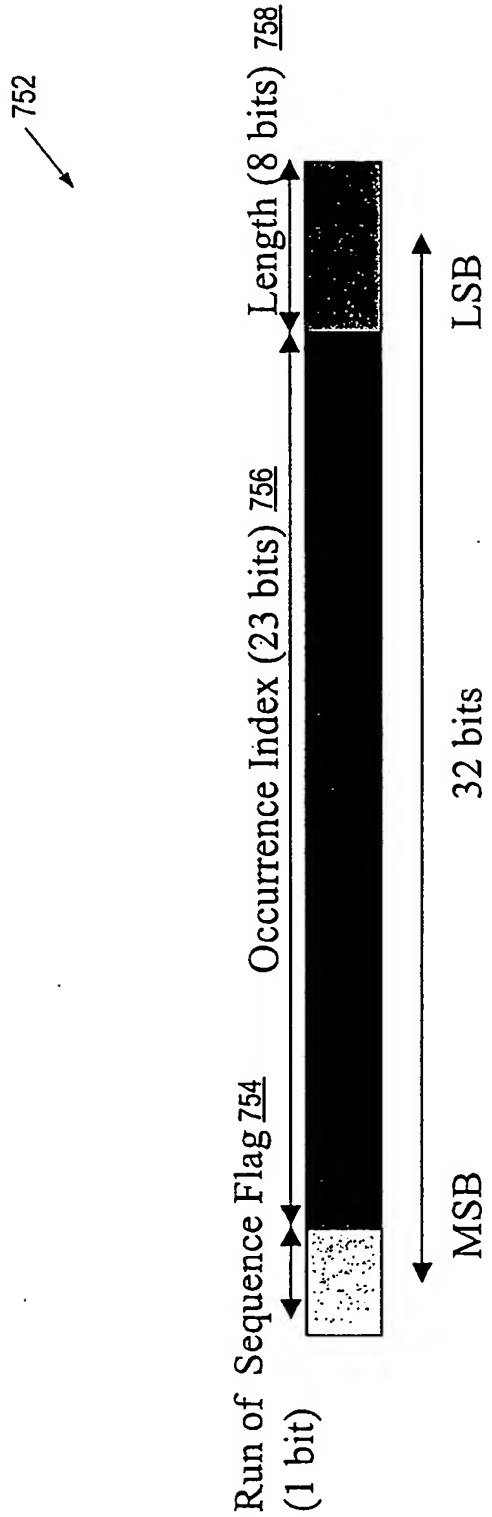
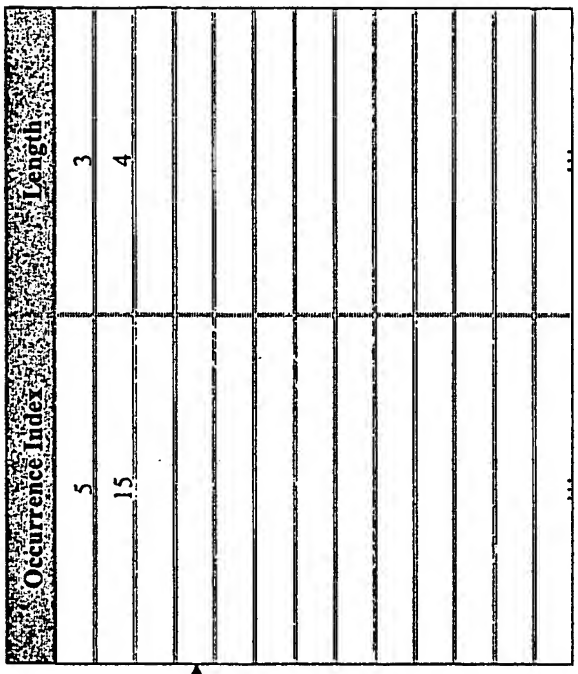

752
Length (8 bits) 758
Occurrence Index (23 bits) 756
Run of Sequence Flag 754
(1 bit)
LSB
32 bits
MSB

FIG. 7J



Repeated Sequence Occurrence Table 760

Structure of a Description ID field.

FIG. 7K

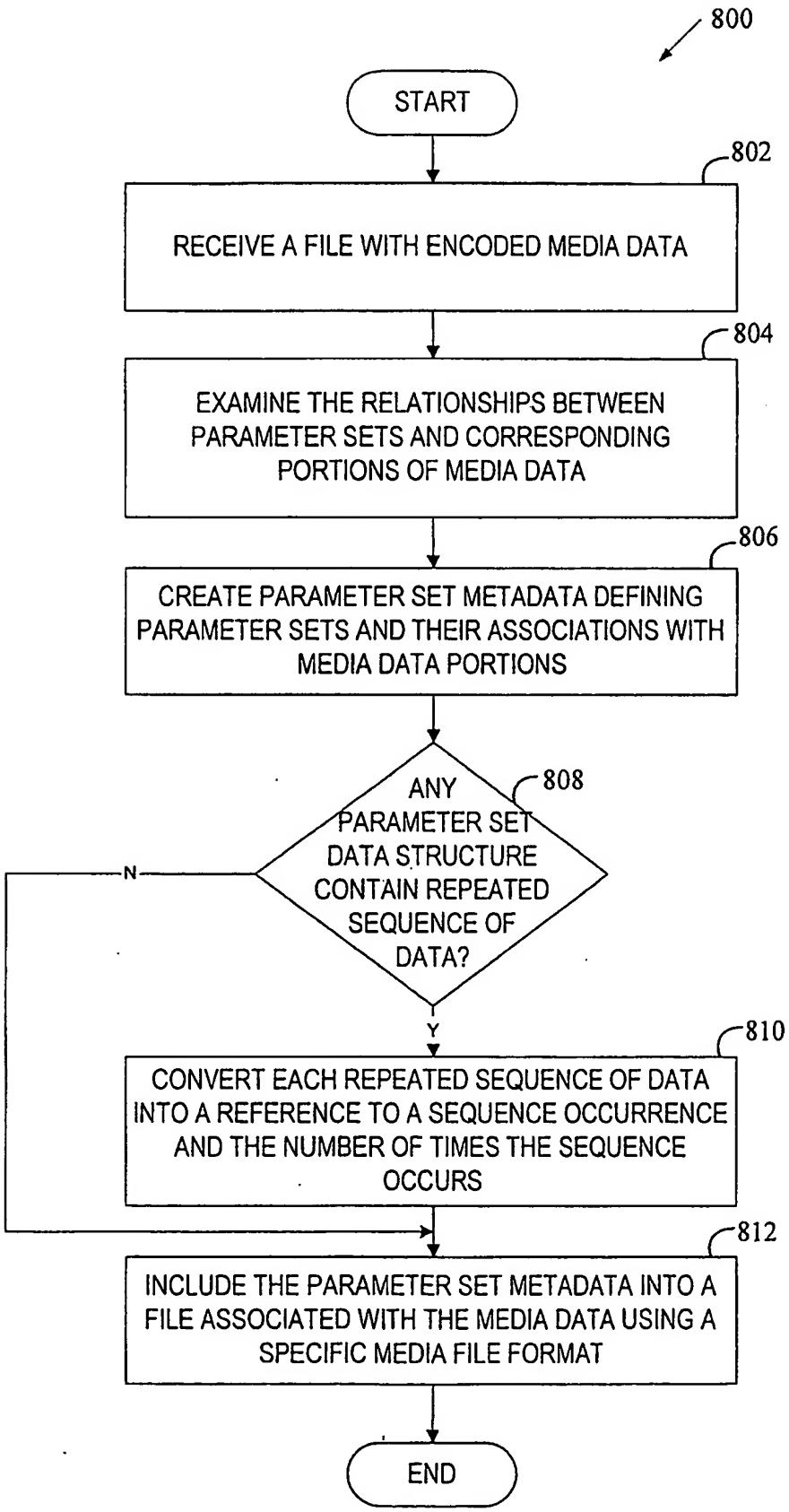

800
START
802
RECEIVE A FILE WITH ENCODED MEDIA DATA
804
EXAMINE THE RELATIONSHIPS BETWEEN PARAMETER SETS AND CORRESPONDING PORTIONS OF MEDIA DATA
806
CREATE PARAMETER SET METADATA DEFINING PARAMETER SETS AND THEIR ASSOCIATIONS WITH MEDIA DATA PORTIONS
808
ANY PARAMETER SET DATA STRUCTURE CONTAIN REPEATED SEQUENCE OF DATA?
N
Y
810
CONVERT EACH REPEATED SEQUENCE OF DATA INTO A REFERENCE TO A SEQUENCE OCCURRENCE AND THE NUMBER OF TIMES THE SEQUENCE OCCURS
812
INCLUDE THE PARAMETER SET METADATA INTO A FILE ASSOCIATED WITH THE MEDIA DATA USING A SPECIFIC MEDIA FILE FORMAT
END

FIG. 8

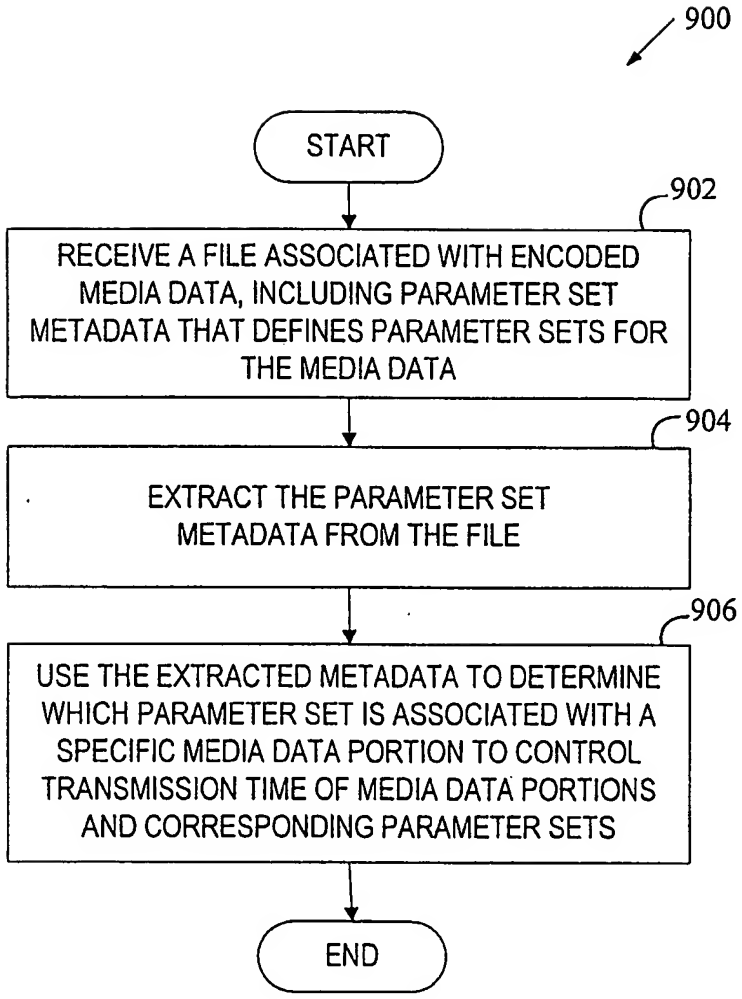

900
START
902
RECEIVE A FILE ASSOCIATED WITH ENCODED MEDIA DATA, INCLUDING PARAMETER SET METADATA THAT DEFINES PARAMETER SETS FOR THE MEDIA DATA
904
EXTRACT THE PARAMETER SET METADATA FROM THE FILE
906
USE THE EXTRACTED METADATA TO DETERMINE WHICH PARAMETER SET IS ASSOCIATED WITH A SPECIFIC MEDIA DATA PORTION TO CONTROL TRANSMISSION TIME OF MEDIA DATA PORTIONS AND CORRESPONDING PARAMETER SETS
END

FIG. 9

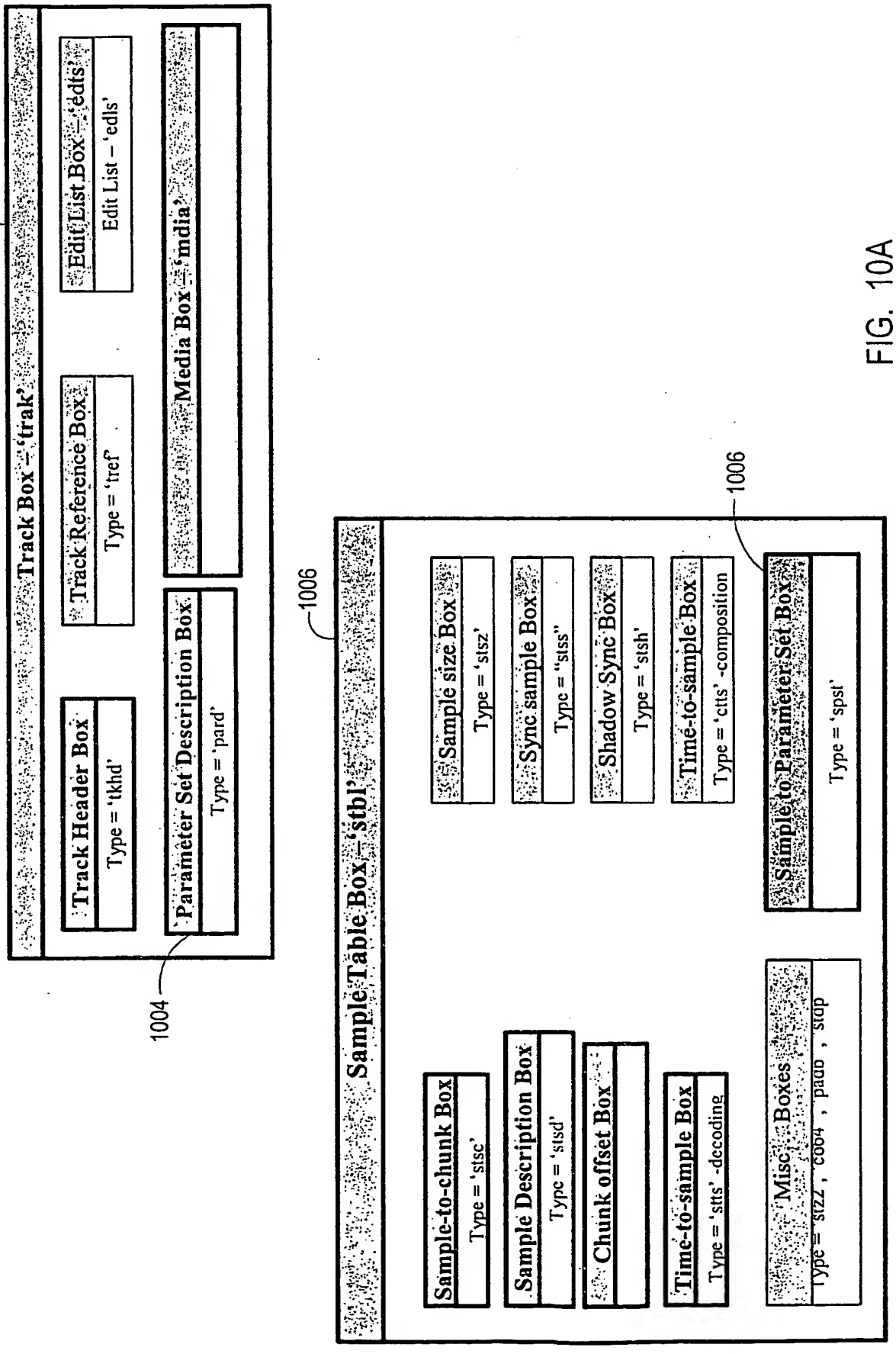

1002
Track Box – 'trak'
Track Header Box
Type = 'tkhd'
Track Reference Box
Type = 'tref'
Edit List Box – 'edts'
Edit List – 'edls'
Parameter Set Description Box
Type = 'pard'
Media Box – 'mdia'
1004
1006
Sample Table Box – 'stbl'
Sample-to-chunk Box
Type = 'stsc'
Sample Description Box
Type = 'stsd'
Chunk offset Box
Time-to-sample Box
Type = 'stts' -decoding
Misc. Boxes
Type = 'stzz', 'co64', 'padb', 'stdp'
Sample size Box
Type = 'stsz'
Sync sample Box
Type = "stss"
Shadow Sync Box
Type = 'stsh'
Time-to-sample Box
Type = 'ctts' -composition
1006
Sample to Parameter Set Box
Type = 'spst'

FIG. 10A

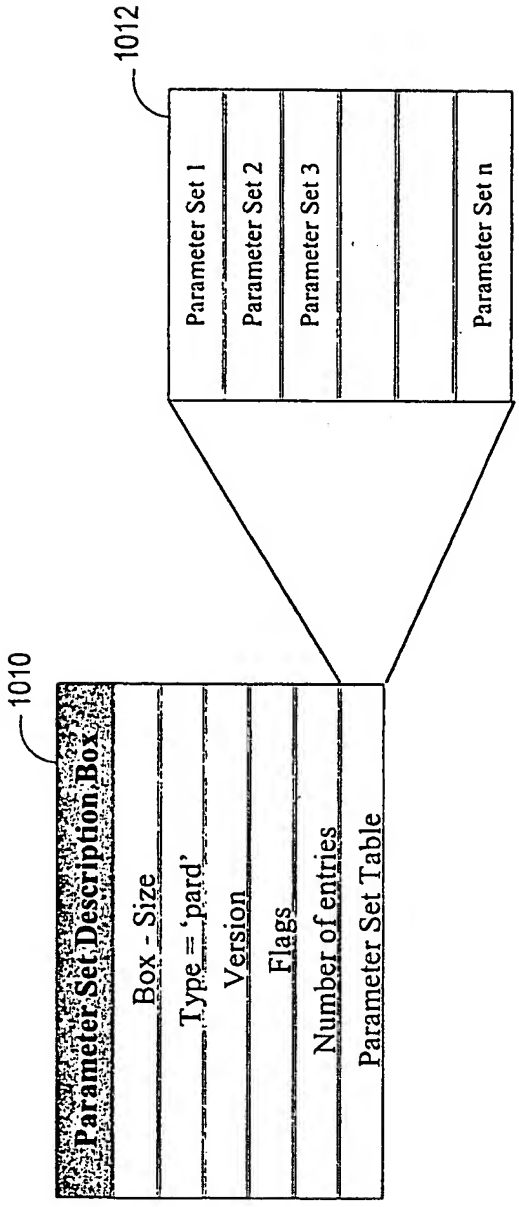

1012
Parameter Set 1
Parameter Set 2
Parameter Set 3
Parameter Set n
1010
Parameter Set Description Box
Box - Size
Type = 'pard'
Version
Flags
Number of entries
Parameter Set Table

FIG. 10B

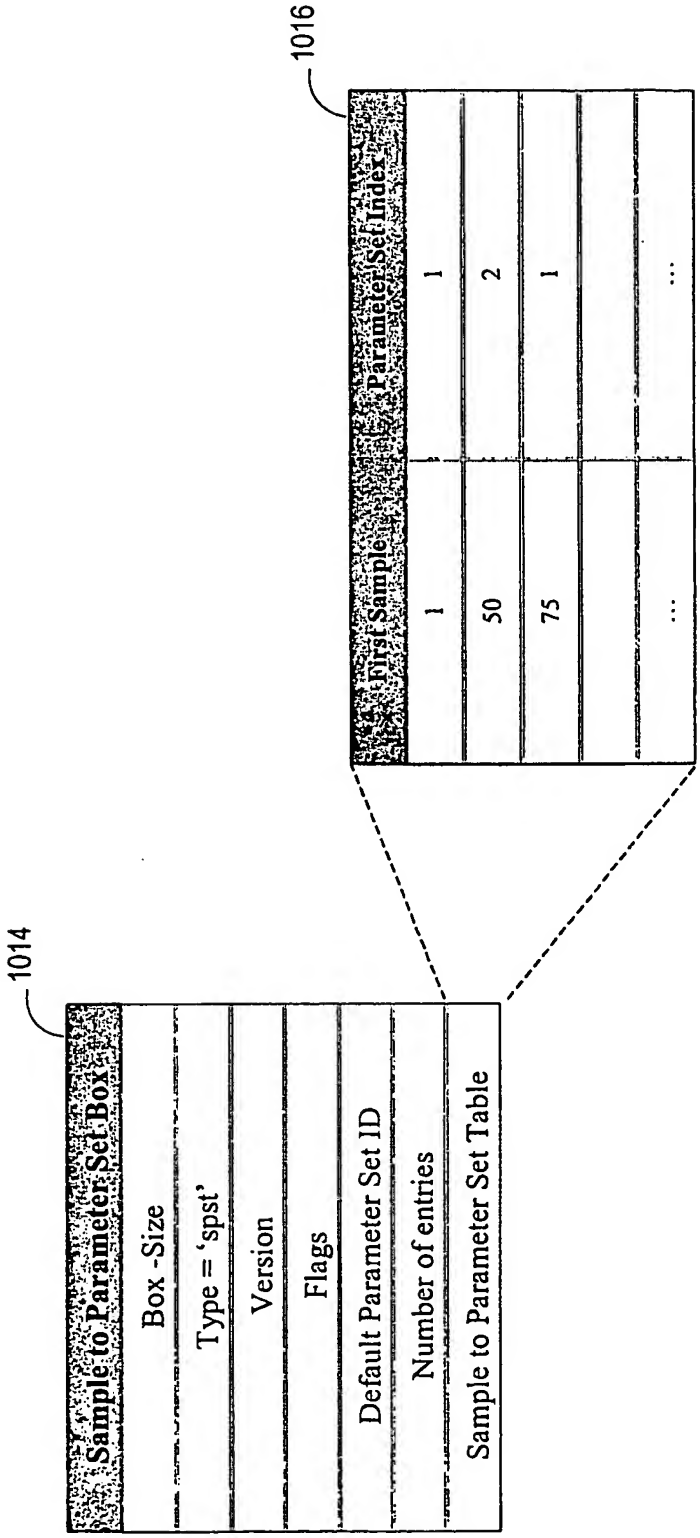

1014
Sample to Parameter Set Box
Box -Size
Type = 'spst'
Version
Flags
Default Parameter Set ID
Number of entries
Sample to Parameter Set Table
1016
First Sample
Parameter Set Index
1
1
50
2
75
1
...
...

FIG. 10C

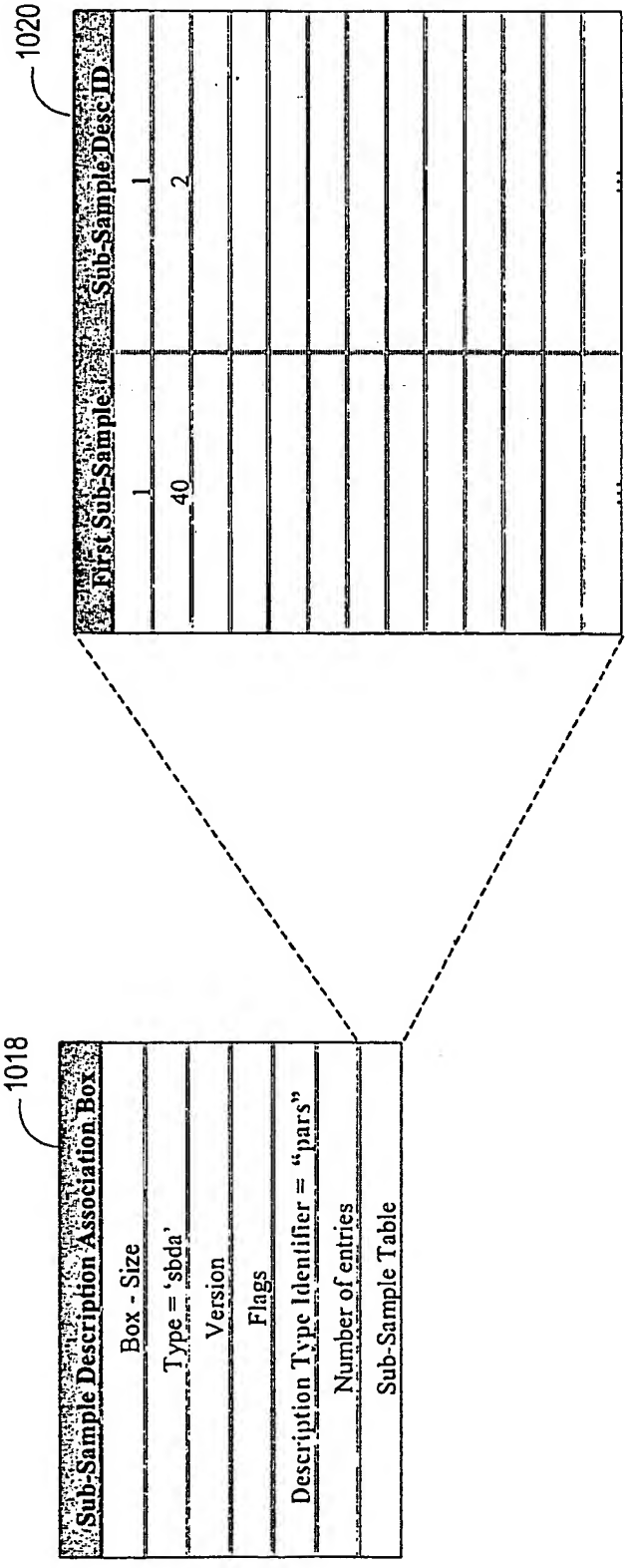

1018
Sub-Sample Description Association Box
Box - Size
Type = 'sbda'
Version
Flags
Description Type Identifier = "pars"
Number of entries
Sub-Sample Table
1020
First Sub-Sample
Sub-Sample Desc ID
1
40
1
2

FIG. 10D

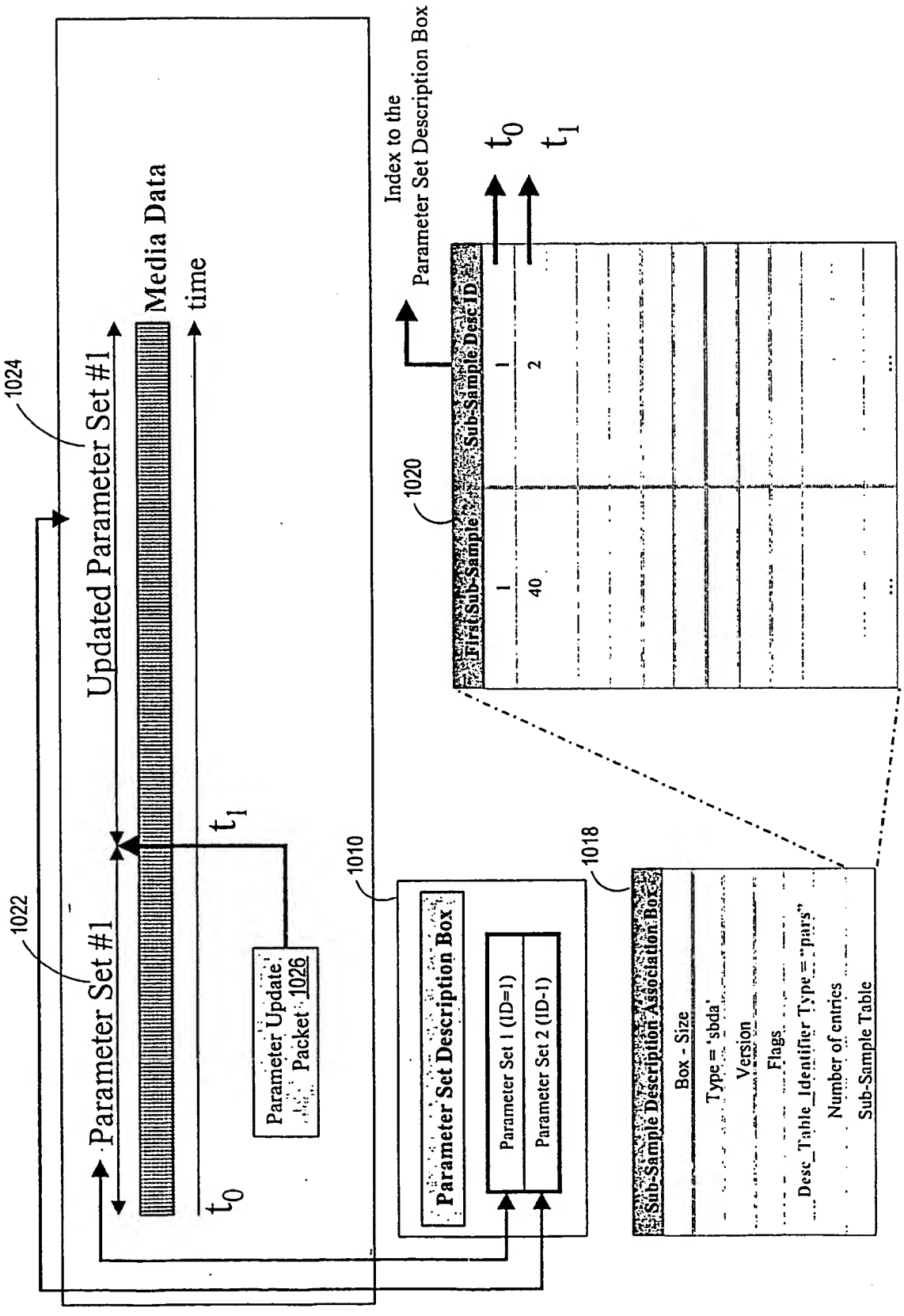

1024
Updated Parameter Set #1
Media Data
time
1022
Parameter Set #1
t_0
t_1
Parameter Update Packet 1026
1010
Parameter Set Description Box
Parameter Set 1 (ID=1)
Parameter Set 2 (ID-1)
1018
Sub-Sample Description Box
Box - Size
Type = 'sbda'
Version
Flags
Desc_Table_Identifier Type = "pars"
Number of entries
Sub-Sample Table
Index to the Parameter Set Description Box
1020
First Sub-Sample
Sub-Sample Desc ID
1
40
1
2
t_0
t_1

FIG. 10E

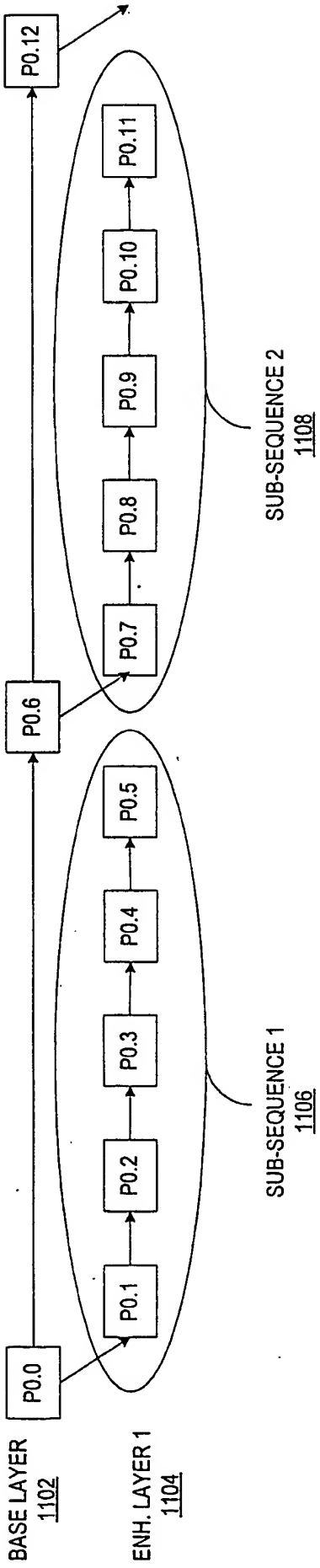

P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P0.8
P0.9
P0.10
P0.11
P0.12
BASE LAYER
1102
ENH. LAYER 1
1104
SUB-SEQUENCE 1
1106
SUB-SEQUENCE 2
1108

FIG. 11

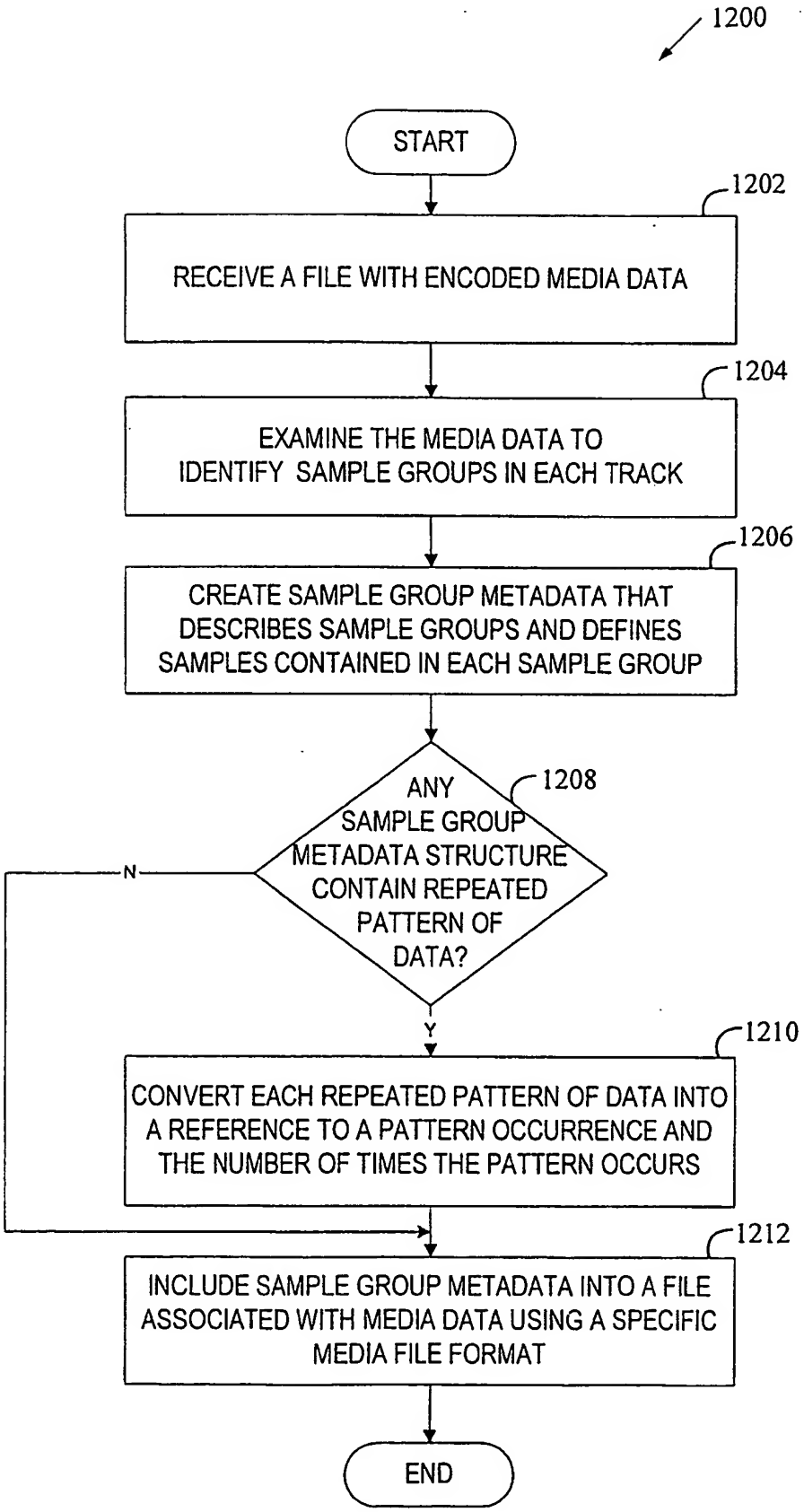

1200
START
1202
RECEIVE A FILE WITH ENCODED MEDIA DATA
1204
EXAMINE THE MEDIA DATA TO IDENTIFY SAMPLE GROUPS IN EACH TRACK
1206
CREATE SAMPLE GROUP METADATA THAT DESCRIBES SAMPLE GROUPS AND DEFINES SAMPLES CONTAINED IN EACH SAMPLE GROUP
1208
ANY SAMPLE GROUP METADATA STRUCTURE CONTAIN REPEATED PATTERN OF DATA?
N
Y
1210
CONVERT EACH REPEATED PATTERN OF DATA INTO A REFERENCE TO A PATTERN OCCURRENCE AND THE NUMBER OF TIMES THE PATTERN OCCURS
1212
INCLUDE SAMPLE GROUP METADATA INTO A FILE ASSOCIATED WITH MEDIA DATA USING A SPECIFIC MEDIA FILE FORMAT
END

FIG. 12

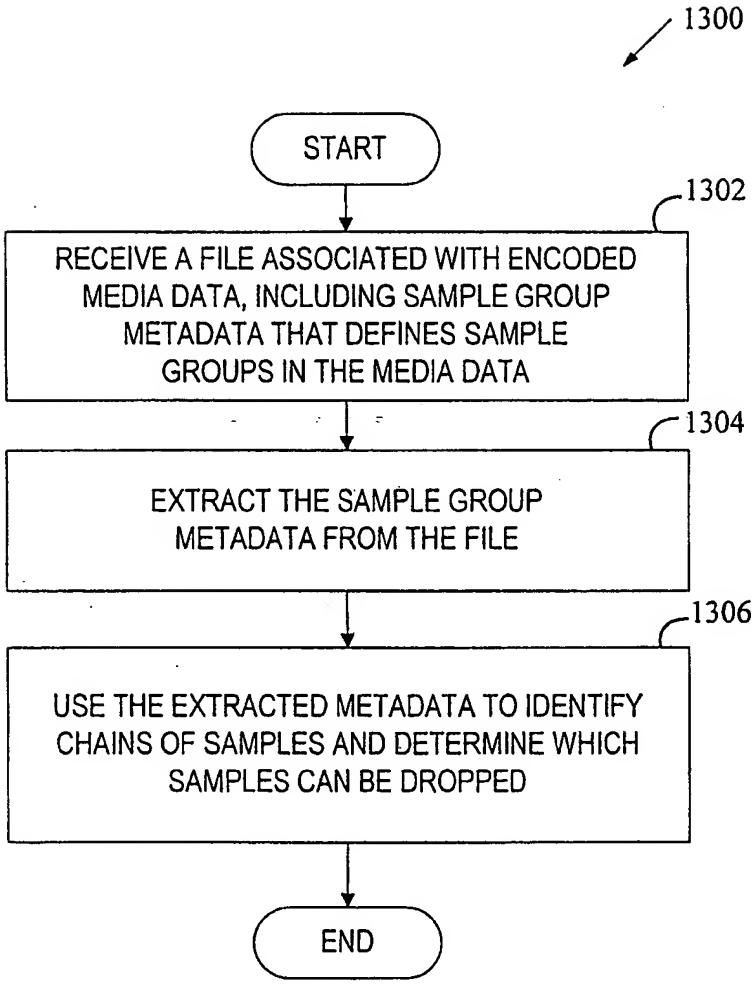

1300
START
1302
RECEIVE A FILE ASSOCIATED WITH ENCODED MEDIA DATA, INCLUDING SAMPLE GROUP METADATA THAT DEFINES SAMPLE GROUPS IN THE MEDIA DATA
1304
EXTRACT THE SAMPLE GROUP METADATA FROM THE FILE
1306
USE THE EXTRACTED METADATA TO IDENTIFY CHAINS OF SAMPLES AND DETERMINE WHICH SAMPLES CAN BE DROPPED
END

FIG. 13

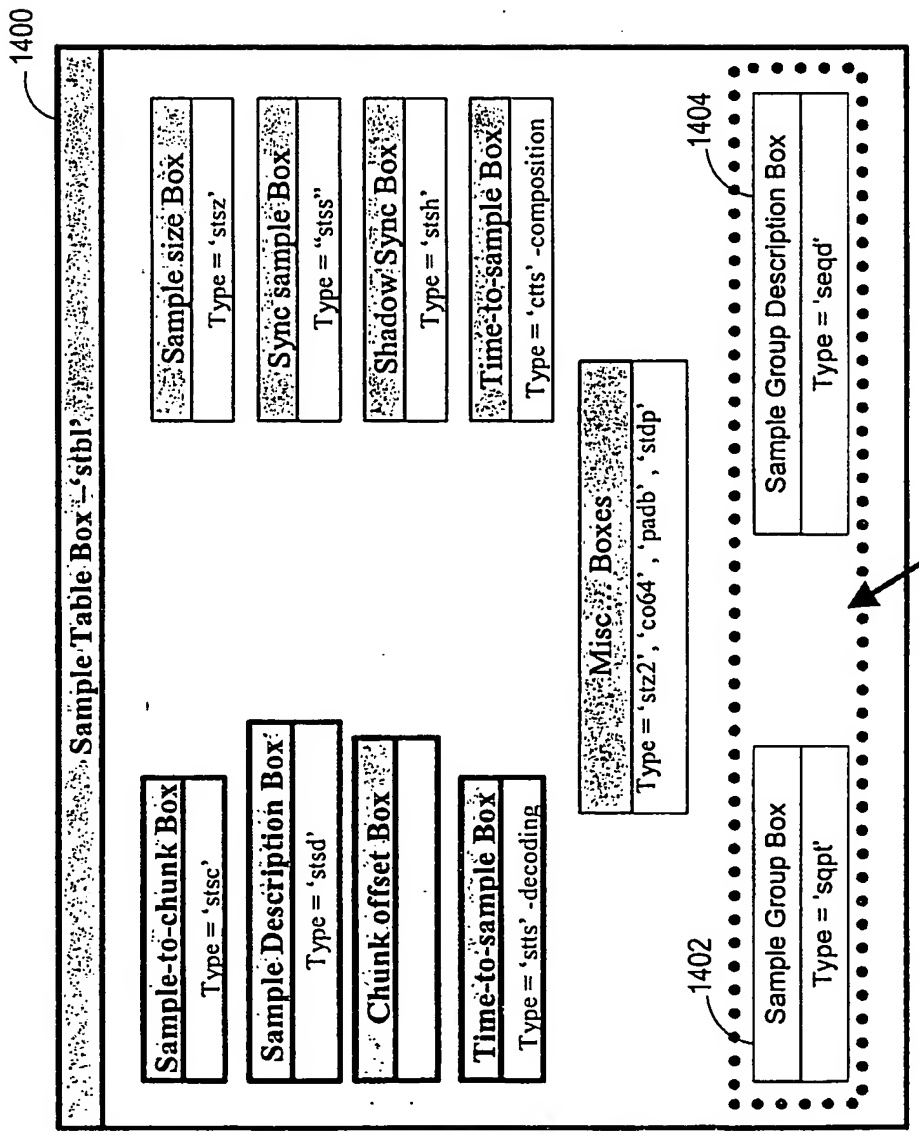

1400
Sample Table Box - 'stbl'
Sample-to-chunk Box
Type = 'stsc'
Sample Description Box
Type = 'stsd'
Chunk offset Box
Time-to-sample Box
Type = 'stts' -decoding
Sample size Box
Type = 'stsz'
Sync sample Box
Type = "stss"
Shadow Sync Box
Type = 'stsh'
Time-to-sample Box
Type = 'ctts' -composition
Misc. Boxes
Type = 'stz2', 'co64', 'padb', 'stdp'
1402
Sample Group Box
Type = 'sqpt'
1404
Sample Group Description Box
Type = 'seqd'
SAMPLE GROUPS

FIG. 14A

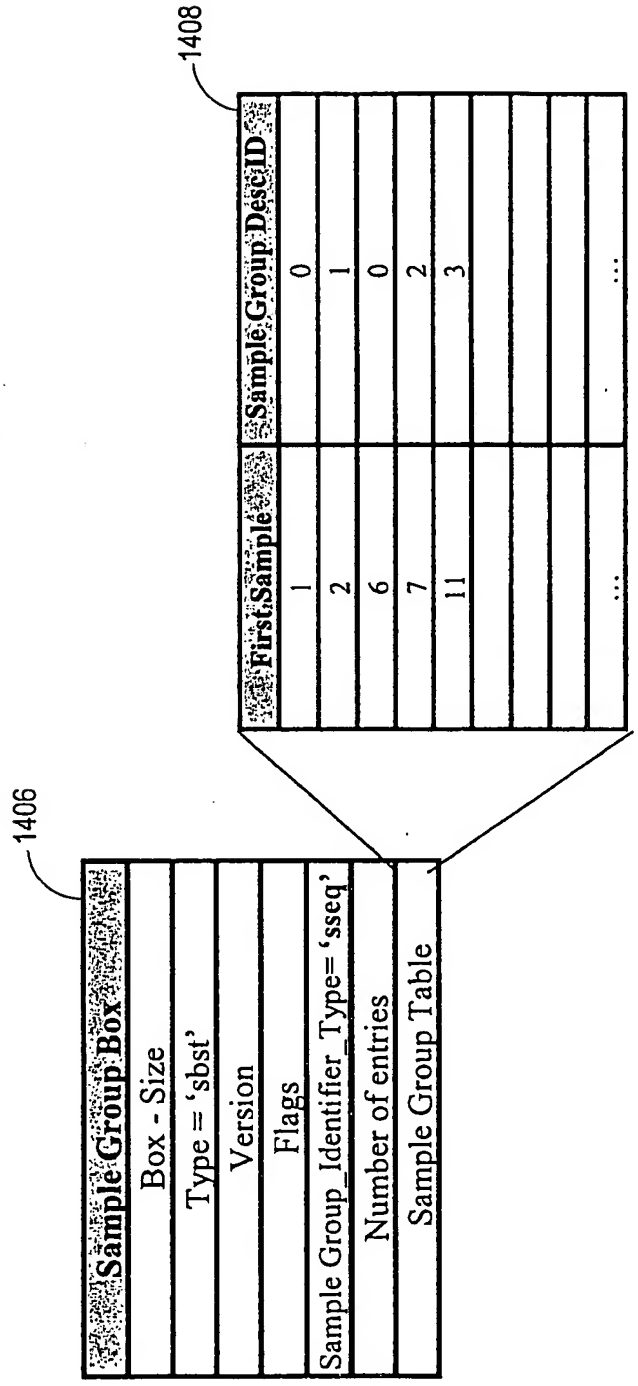

1406
Sample Group Box
Box - Size
Type = 'sbst'
Version
Flags
Sample Group_Identifier_Type= 'sseq'
Number of entries
Sample Group Table
1408
First Sample
Sample Group DescID
1
0
2
1
6
0
7
2
11
3
...
...

FIG. 14B

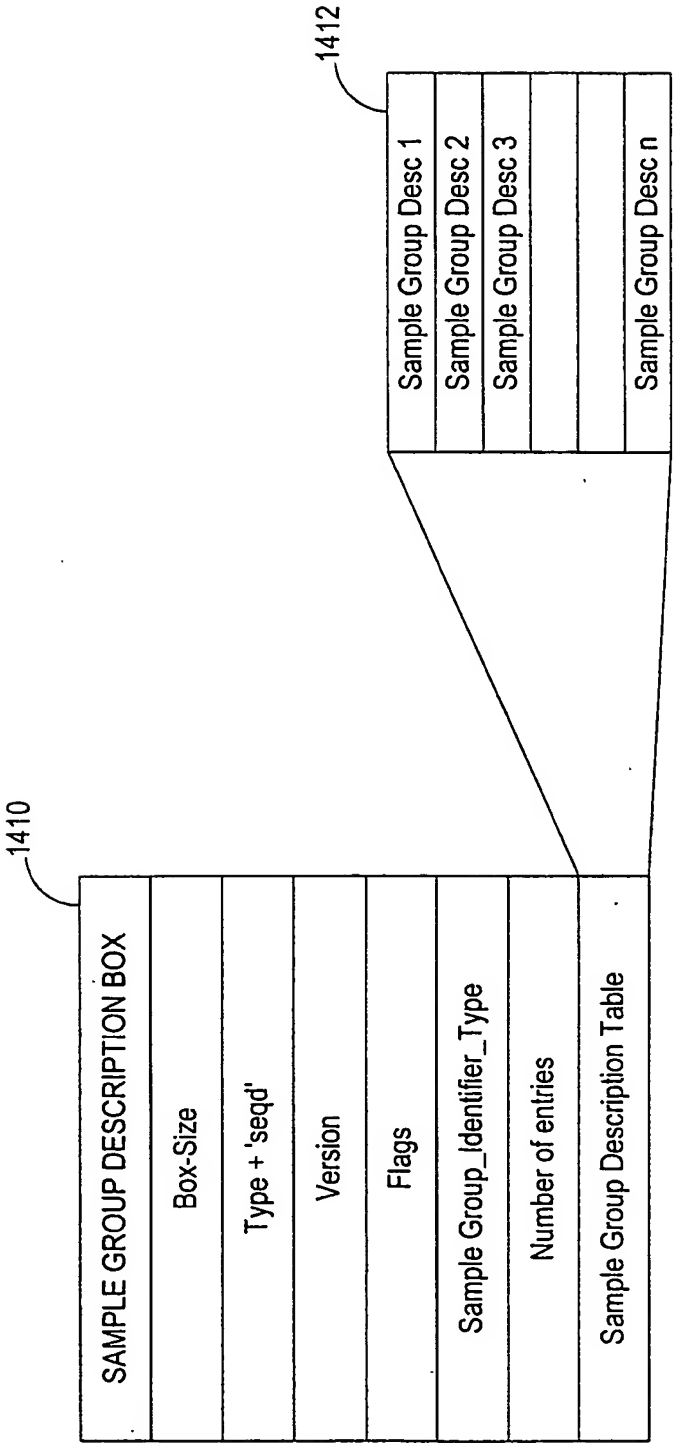

1410
SAMPLE GROUP DESCRIPTION BOX
Box-Size
Type + 'seqd'
Version
Flags
Sample Group_Identifier_Type
Number of entries
Sample Group Description Table
1412
Sample Group Desc 1
Sample Group Desc 2
Sample Group Desc 3
Sample Group Desc n

FIG. 14C

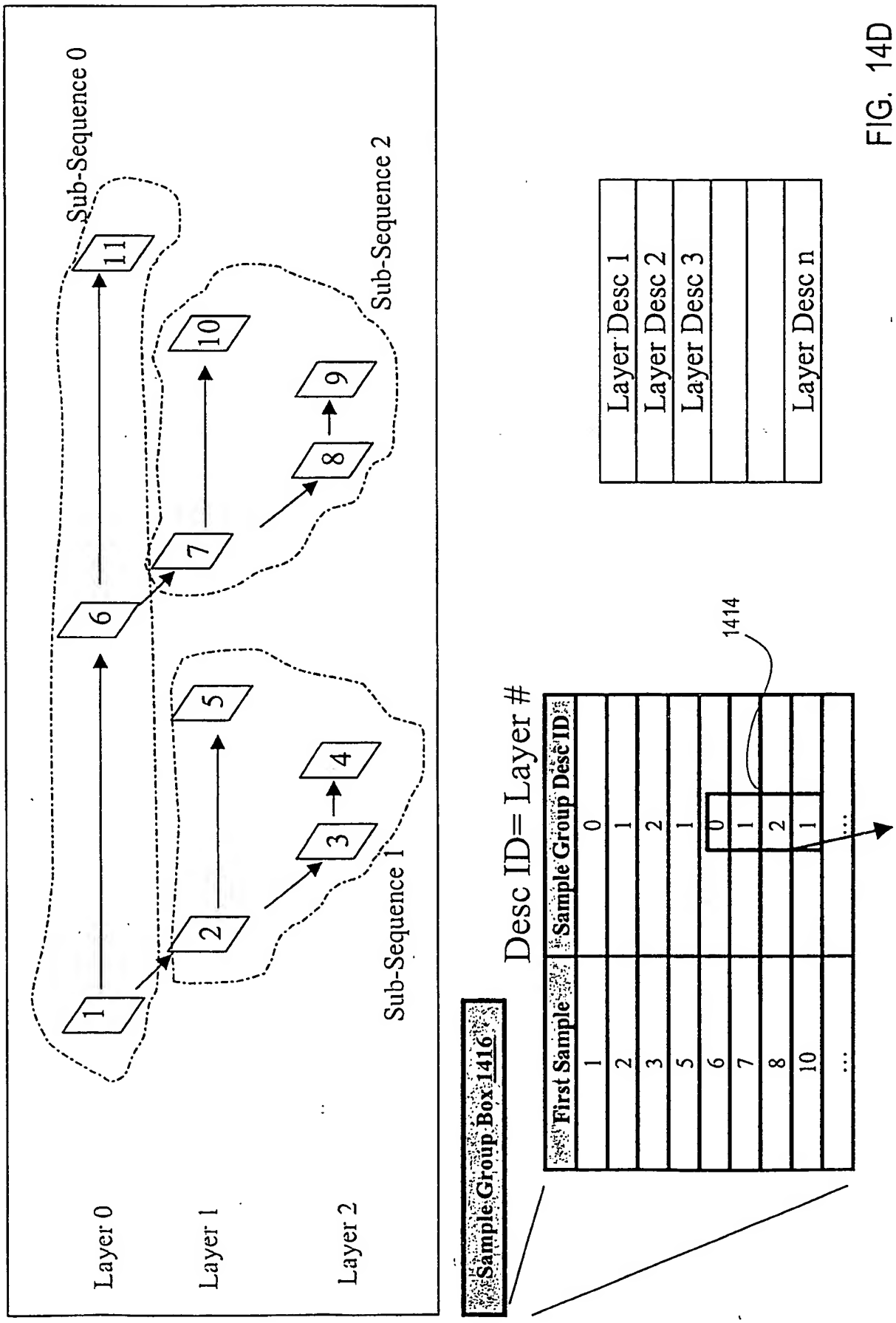

Sub-Sequence 0
Sub-Sequence 1
Sub-Sequence 2
Layer 0
Layer 1
Layer 2
1
2
3
4
5
6
7
8
9
10
11
Sample Group Box 1416
Desc ID= Layer #
First Sample
Sample Group Desc ID
1414
Layer Desc 1
Layer Desc 2
Layer Desc 3
Layer Desc n

FIG. 14D

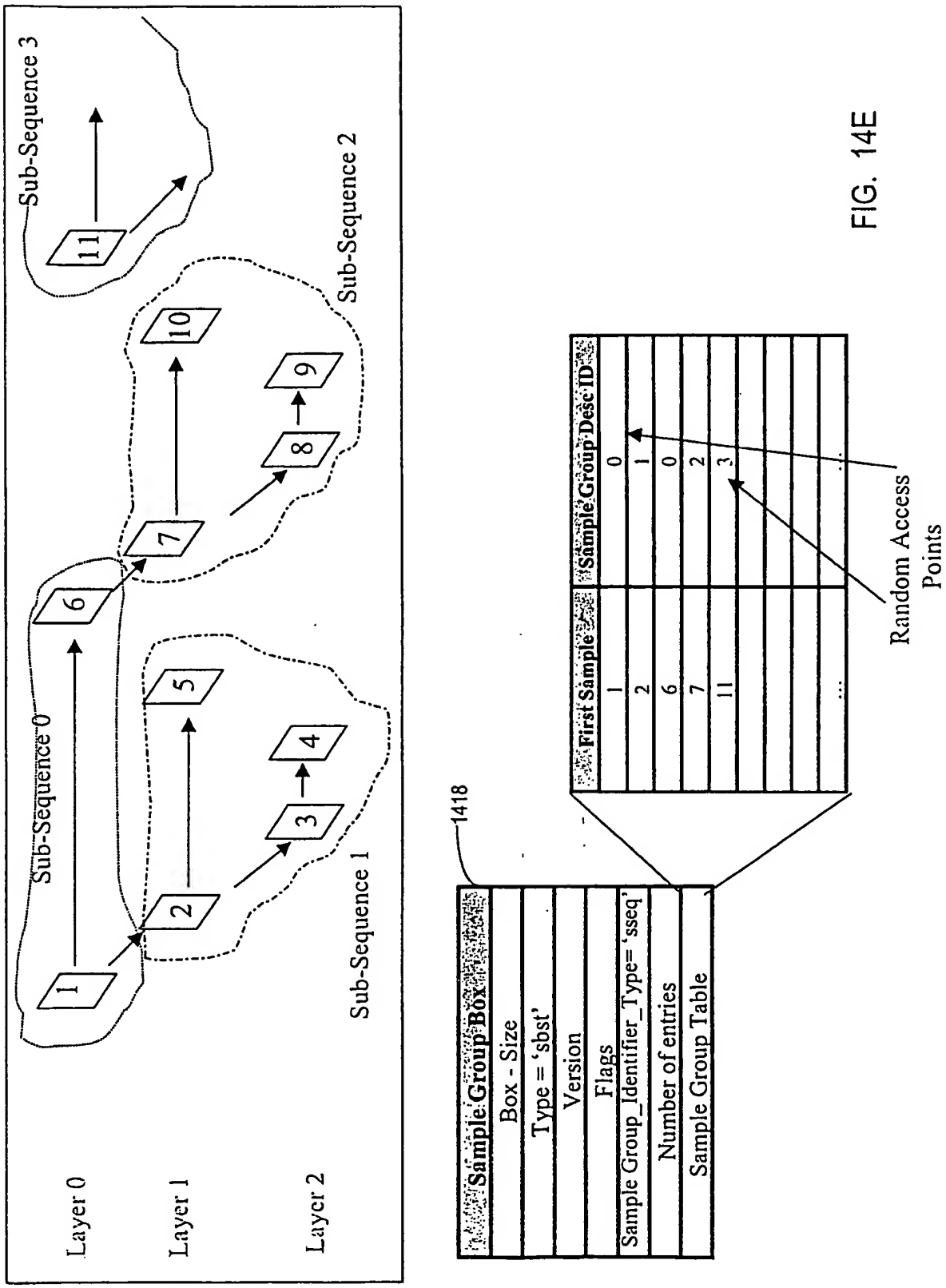

Layer 0
Layer 1
Layer 2
Sub-Sequence 0
Sub-Sequence 1
Sub-Sequence 2
Sub-Sequence 3
1
2
3
4
5
6
7
8
9
10
11
1418
Sample Group Box
Box - Size
Type = 'sbst'
Version
Flags
Sample Group_Identifier_Type= 'sseq'
Number of entries
Sample Group Table
First Sample
Sample Group Desc ID
1
0
2
1
6
0
7
2
11
3
...
...
Random Access Points

FIG. 14E

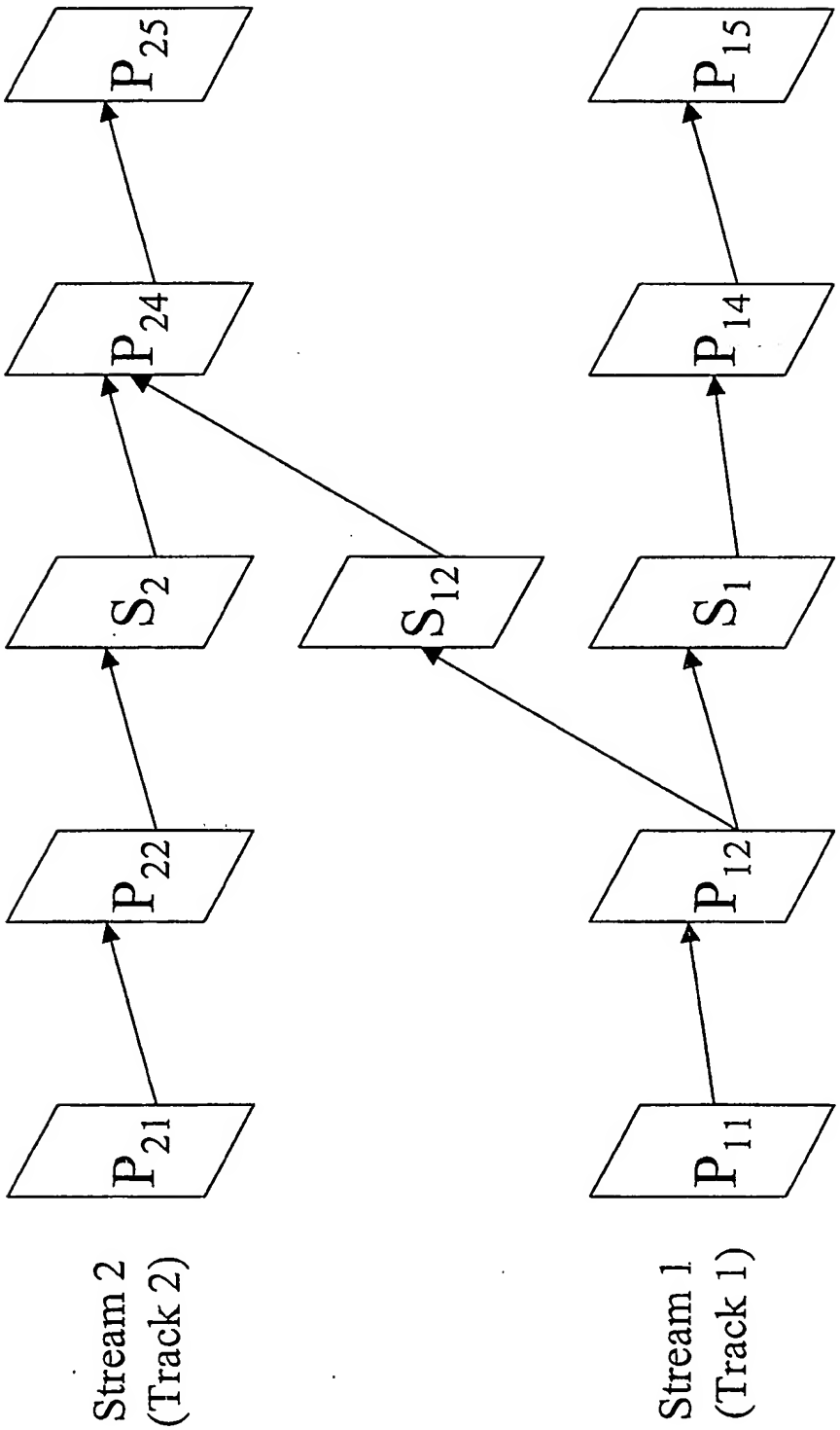

Stream 2
(Track 2)
Stream 1
(Track 1)
P21
P22
S2
P24
P25
S12
P11
P12
S1
P14
P15

FIG. 15A

Switch Sample Set 1502

Switch Sample	*Switch Sample Track*	*Reference Track*	*Reference Samples*
S_2	2	2	P_{22}
S_{12}	XXX	1	P_{12}

FIG. 15B

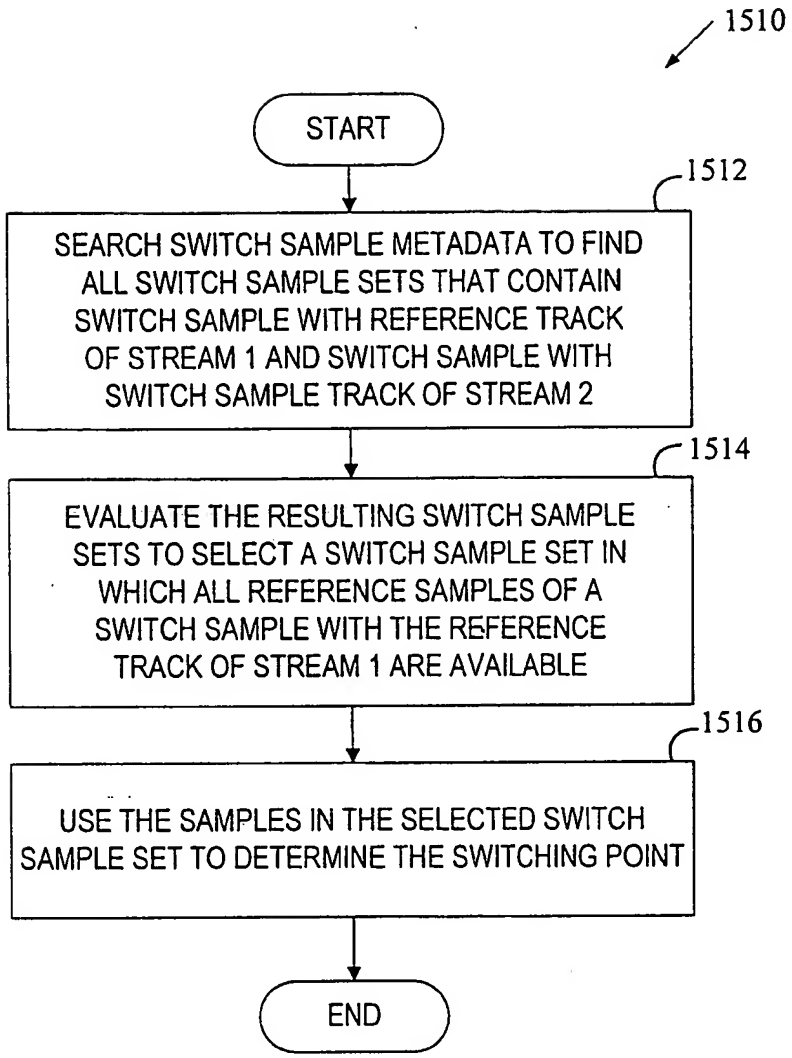

1510
START
1512
SEARCH SWITCH SAMPLE METADATA TO FIND ALL SWITCH SAMPLE SETS THAT CONTAIN SWITCH SAMPLE WITH REFERENCE TRACK OF STREAM 1 AND SWITCH SAMPLE WITH SWITCH SAMPLE TRACK OF STREAM 2
1514
EVALUATE THE RESULTING SWITCH SAMPLE SETS TO SELECT A SWITCH SAMPLE SET IN WHICH ALL REFERENCE SAMPLES OF A SWITCH SAMPLE WITH THE REFERENCE TRACK OF STREAM 1 ARE AVAILABLE
1516
USE THE SAMPLES IN THE SELECTED SWITCH SAMPLE SET TO DETERMINE THE SWITCHING POINT
END

FIG. 15C

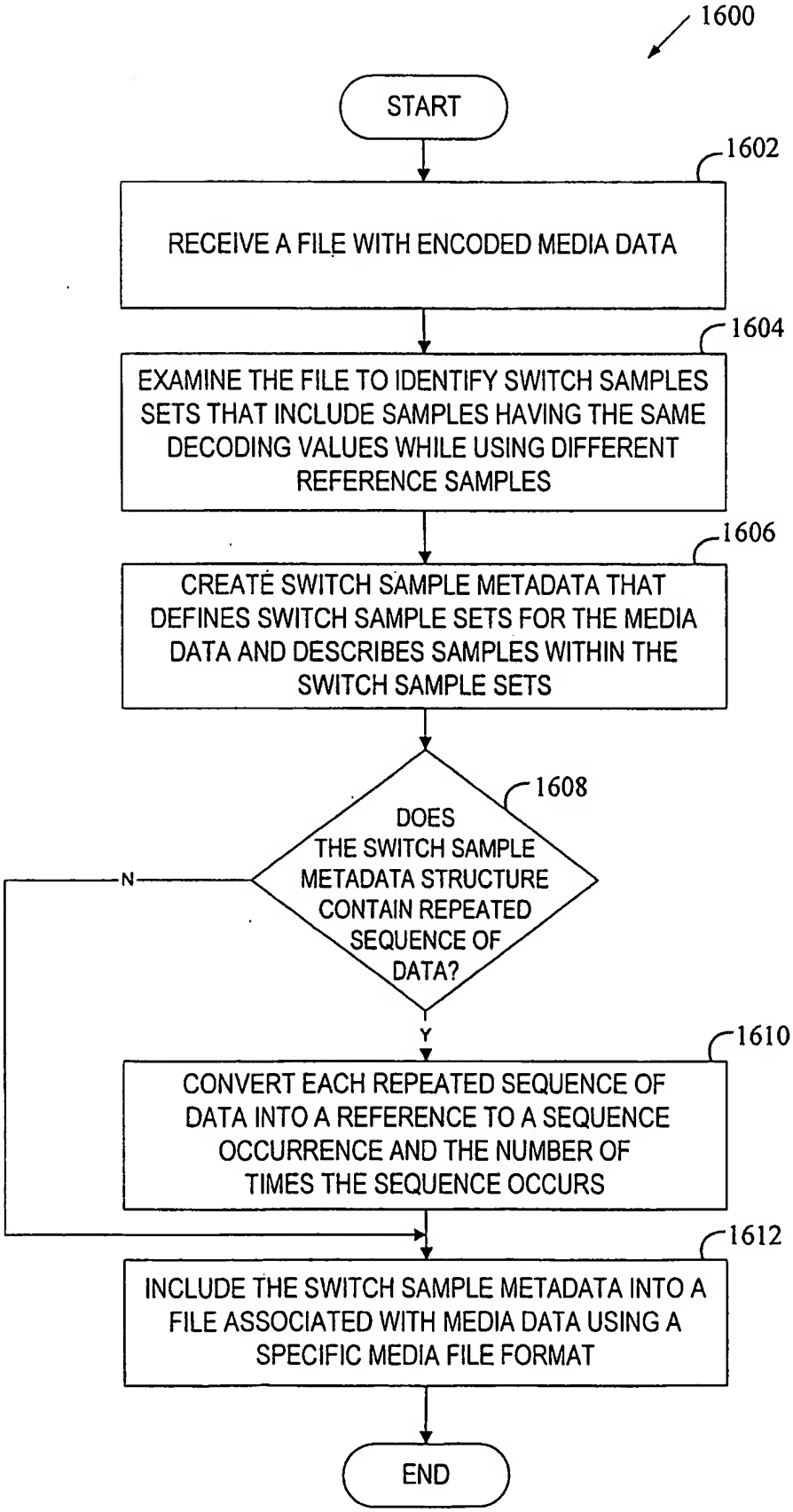

1600
START
1602
RECEIVE A FILE WITH ENCODED MEDIA DATA
1604
EXAMINE THE FILE TO IDENTIFY SWITCH SAMPLES SETS THAT INCLUDE SAMPLES HAVING THE SAME DECODING VALUES WHILE USING DIFFERENT REFERENCE SAMPLES
1606
CREATE SWITCH SAMPLE METADATA THAT DEFINES SWITCH SAMPLE SETS FOR THE MEDIA DATA AND DESCRIBES SAMPLES WITHIN THE SWITCH SAMPLE SETS
1608
DOES THE SWITCH SAMPLE METADATA STRUCTURE CONTAIN REPEATED SEQUENCE OF DATA?
N
Y
1610
CONVERT EACH REPEATED SEQUENCE OF DATA INTO A REFERENCE TO A SEQUENCE OCCURRENCE AND THE NUMBER OF TIMES THE SEQUENCE OCCURS
1612
INCLUDE THE SWITCH SAMPLE METADATA INTO A FILE ASSOCIATED WITH MEDIA DATA USING A SPECIFIC MEDIA FILE FORMAT
END

FIG. 16

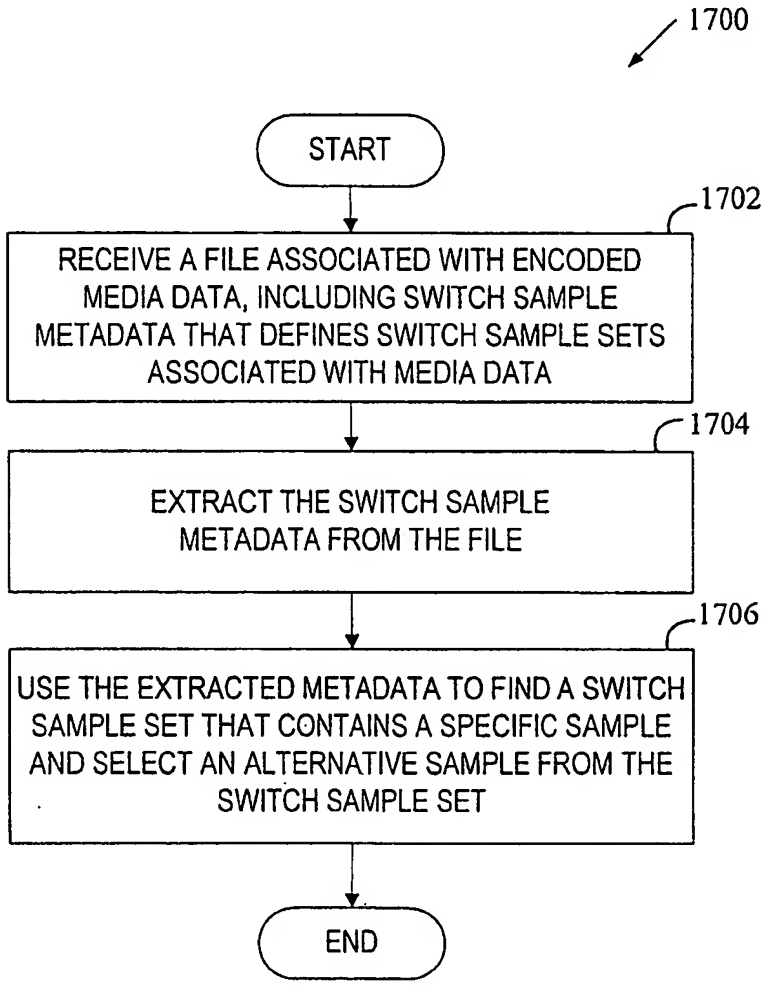

1700
START
1702
RECEIVE A FILE ASSOCIATED WITH ENCODED MEDIA DATA, INCLUDING SWITCH SAMPLE METADATA THAT DEFINES SWITCH SAMPLE SETS ASSOCIATED WITH MEDIA DATA
1704
EXTRACT THE SWITCH SAMPLE METADATA FROM THE FILE
1706
USE THE EXTRACTED METADATA TO FIND A SWITCH SAMPLE SET THAT CONTAINS A SPECIFIC SAMPLE AND SELECT AN ALTERNATIVE SAMPLE FROM THE SWITCH SAMPLE SET
END

FIG. 17

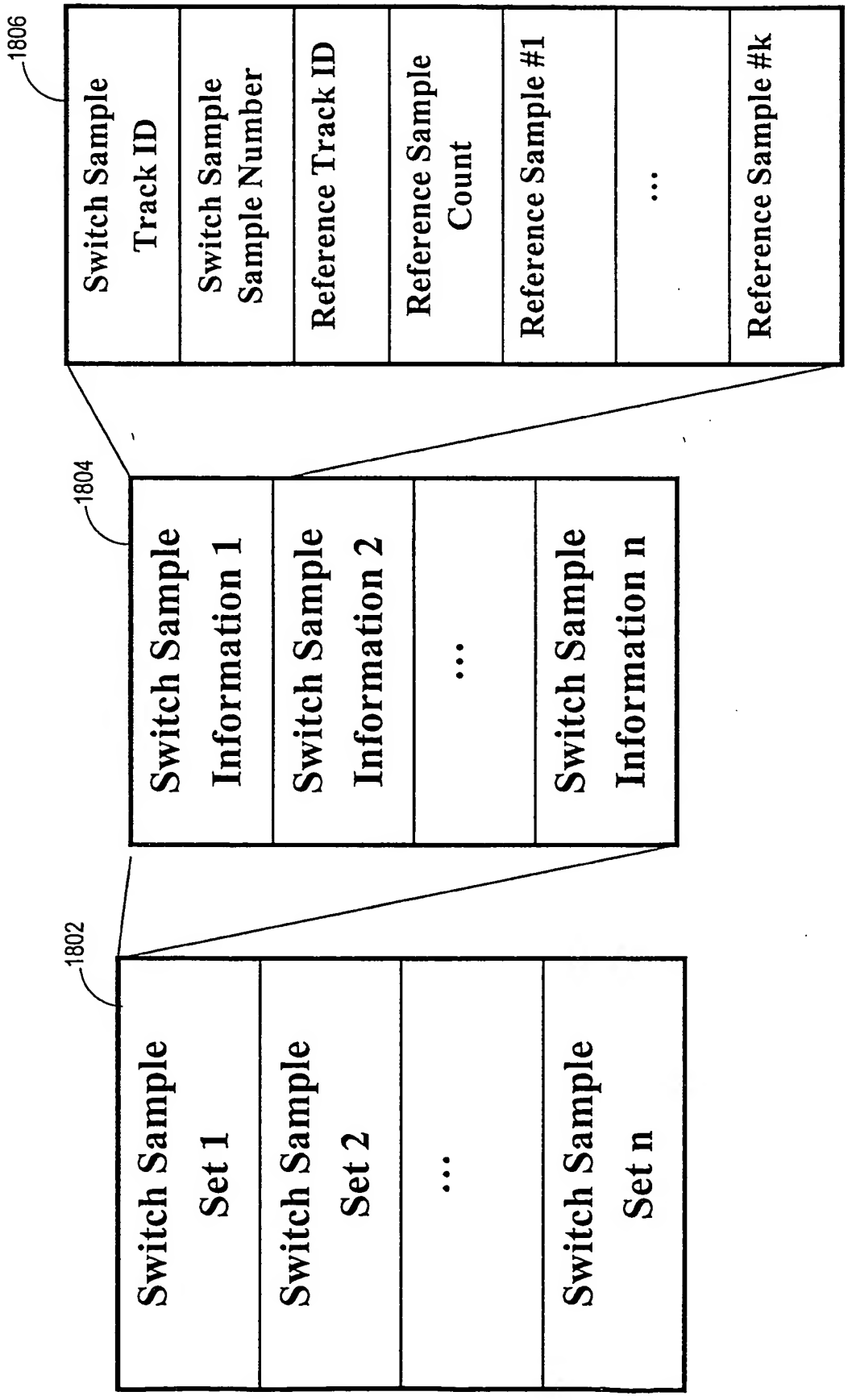

1806
Switch Sample Track ID
Switch Sample Sample Number
Reference Track ID
Reference Sample Count
Reference Sample #1
...
Reference Sample #k
1804
Switch Sample Information 1
Switch Sample Information 2
...
Switch Sample Information n
1802
Switch Sample Set 1
Switch Sample Set 2
...
Switch Sample Set n

FIG. 18

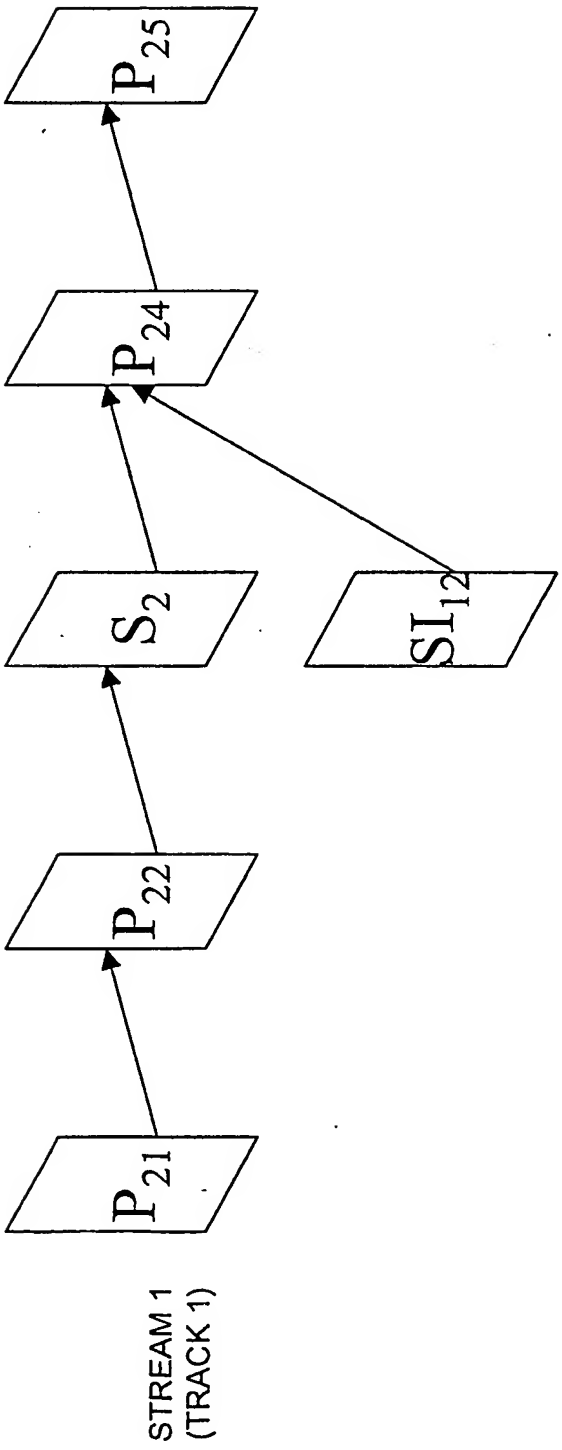


FIG. 19A

Switch Sample Set 1902

Switch Sample	*Switch Sample Track*	*Reference Track*	*Reference Samples*
S_2	1	1	P_{22}
S_{12}	xxx	1	N/A

FIG. 19B

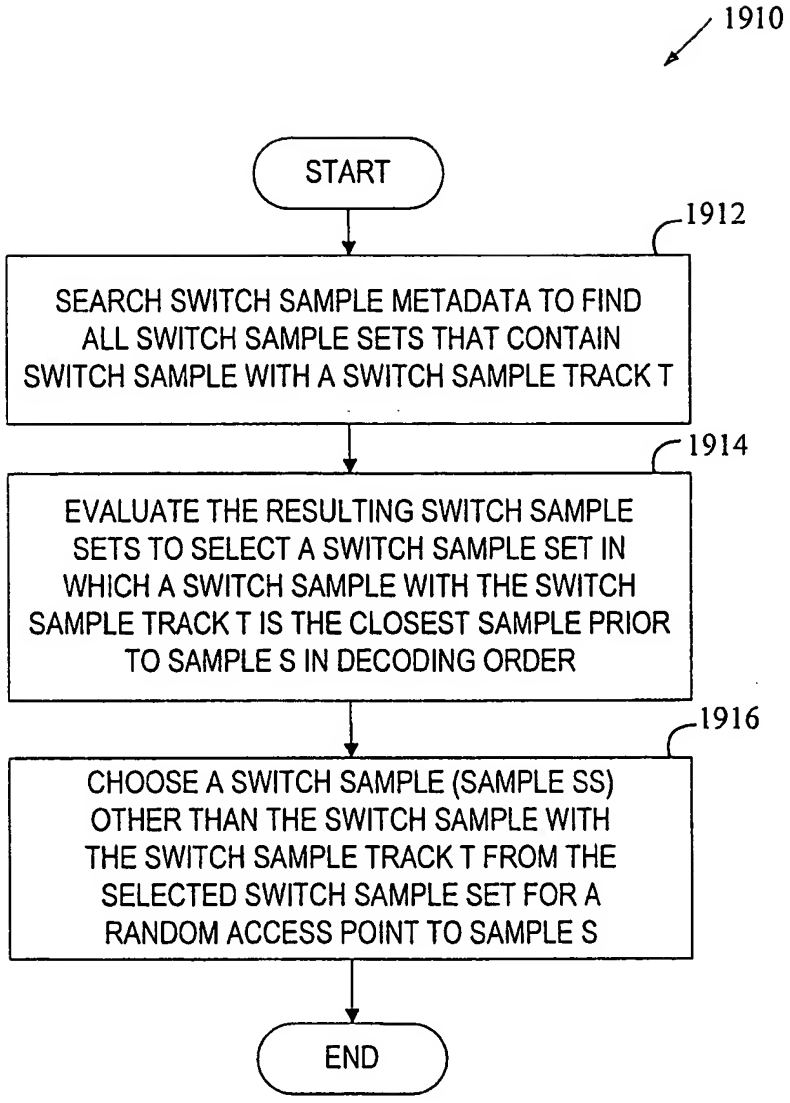

1910
START
1912
SEARCH SWITCH SAMPLE METADATA TO FIND ALL SWITCH SAMPLE SETS THAT CONTAIN SWITCH SAMPLE WITH A SWITCH SAMPLE TRACK T
1914
EVALUATE THE RESULTING SWITCH SAMPLE SETS TO SELECT A SWITCH SAMPLE SET IN WHICH A SWITCH SAMPLE WITH THE SWITCH SAMPLE TRACK T IS THE CLOSEST SAMPLE PRIOR TO SAMPLE S IN DECODING ORDER
1916
CHOOSE A SWITCH SAMPLE (SAMPLE SS) OTHER THAN THE SWITCH SAMPLE WITH THE SWITCH SAMPLE TRACK T FROM THE SELECTED SWITCH SAMPLE SET FOR A RANDOM ACCESS POINT TO SAMPLE S
END

FIG. 19C

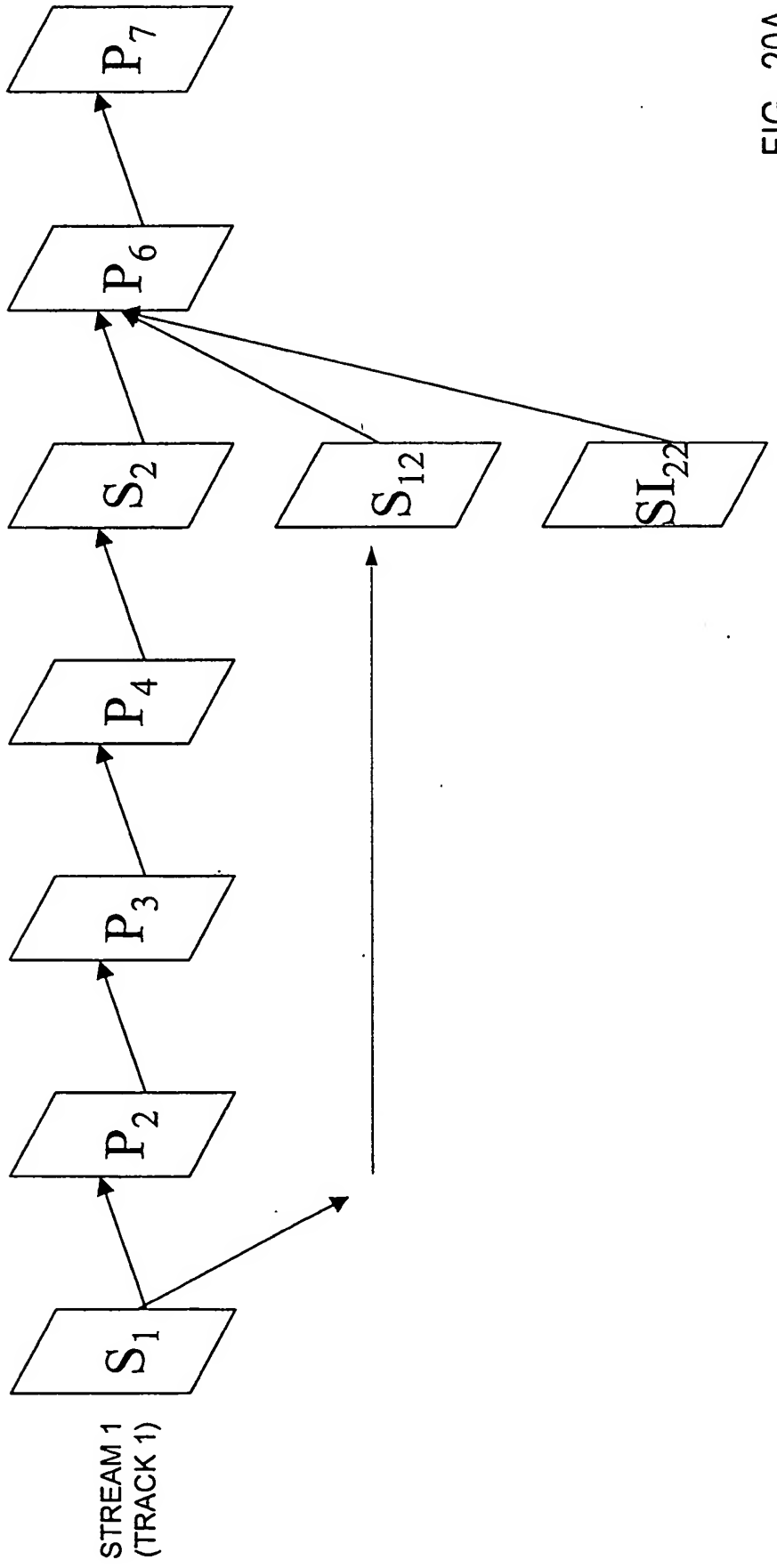
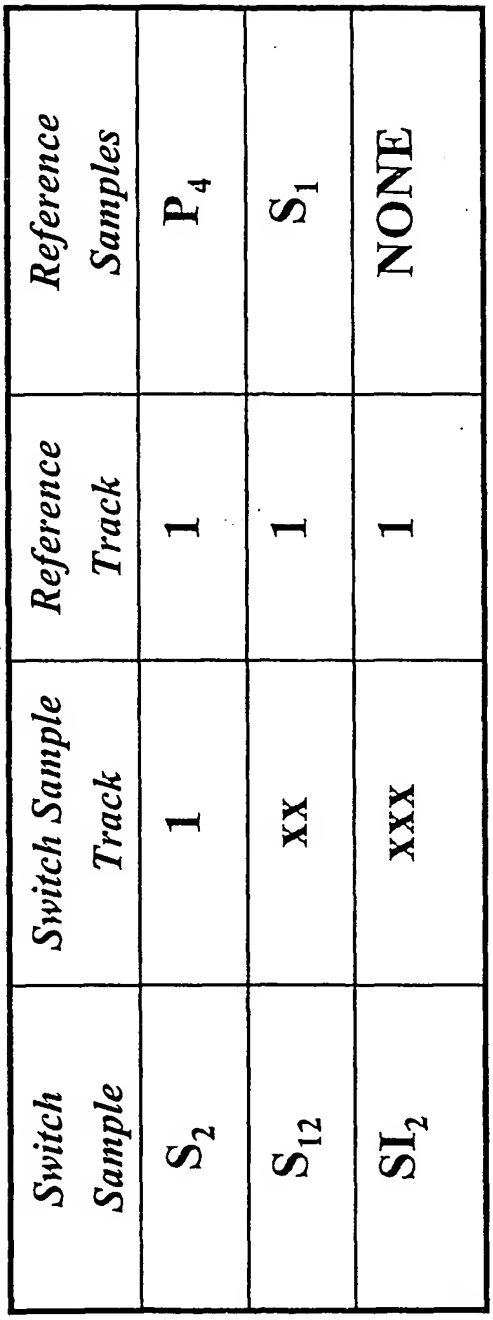


FIG. 20A



Switch Sample	*Switch Sample Track*	*Reference Track*	*Reference Samples*
S_2	1	1	P_4
S_{12}	XX	1	S_1
SI_2	XXX	1	NONE

Switch Sample Set 2002

FIG. 20B

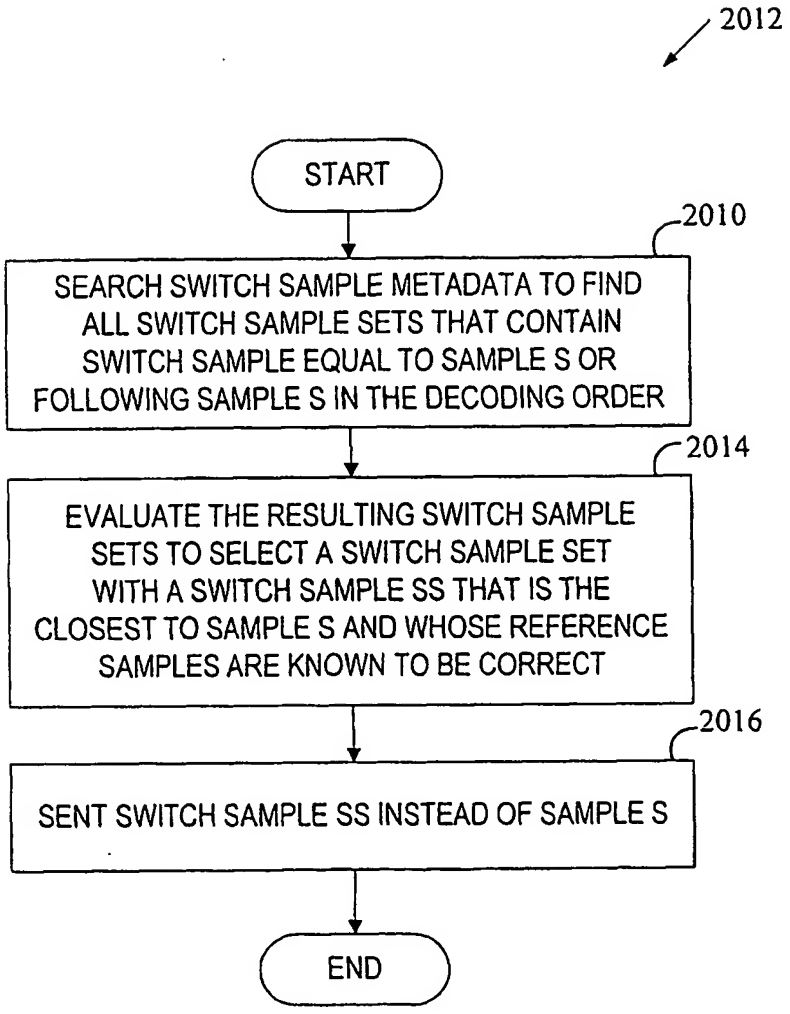

2012
START
2010
SEARCH SWITCH SAMPLE METADATA TO FIND ALL SWITCH SAMPLE SETS THAT CONTAIN SWITCH SAMPLE EQUAL TO SAMPLE S OR FOLLOWING SAMPLE S IN THE DECODING ORDER
2014
EVALUATE THE RESULTING SWITCH SAMPLE SETS TO SELECT A SWITCH SAMPLE SET WITH A SWITCH SAMPLE SS THAT IS THE CLOSEST TO SAMPLE S AND WHOSE REFERENCE SAMPLES ARE KNOWN TO BE CORRECT
2016
SENT SWITCH SAMPLE SS INSTEAD OF SAMPLE S
END

FIG. 20C

INTERNATIONAL SEARCH REPORT

Inter ›nal Application No
PCT/US 03/05636

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	WALKER T ET AL: "FIRST IDEAS ON THE STORAGE OF AVC CONTENT IN MP4" INTERNATIONAL STANDARD ISO/IEC, XX, XX, May 2002 (2002-05), pages 1-39, XP001089885 the whole document --- -/--	1-85

[X] Further documents are listed in the continuation of box C.

[ ] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 June 2003	10/07/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Fantini, F

INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/US 03/05636

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	HANNUKSELA M M: "H.26L FILE FORMAT" ITU TELECOMMUNICATIONS STANDARDIZATION SECTOR VCEG-044, XX, XX, 28 November 2001 (2001-11-28), page COMPLETE XP001148202	1-5, 7-14, 58-61, 65-71, 78,79
Y	paragraph '0001!	15-33, 35-44, 46-49, 51-56, 62-64, 72-77, 80-85
	paragraph '02.1! paragraph '02.2! paragraph '03.1! paragraph '4.3.1! paragraph '4.4.1!	
A	paragraph '4.8.1!	6,34,45, 50,57

Y	M. M. HANNUKSELA: "Enhanced Concept of GOP" ISO/IEC JTC1/SC29/WG11 AND ITU-T SG16 Q.6 (JVT-B042), 23 January 2002 (2002-01-23), XP002245421	15-33, 35-44, 46-49, 51-56, 62-64, 72-77, 80-85
	paragraph '1.2.1! paragraph '1.2.2! paragraph '1.3.2! paragraph '02.1!	
A	paragraph '02.2!	1-14,34, 45,50, 57-61, 65-71, 78,79

	-/--	

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 03/05636

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	WENGER S ET AL: "RTP payload Format for JVT Video" RTP PAYLOAD FORMAT FOR JVT VIDEO, 21 February 2002 (2002-02-21), XP002227245 Retrieved from the Internet: <URL:http://www.cs.columbia.edu/{hgs/rtp/drafts/draft-wenger-avt-rtp-jvt-00.t> 'retrieved on 2003-01-14! paragraph '0001! paragraph '03.1! paragraph '03.2! paragraph '0004! paragraph '05.1! paragraph '0006!	1-5, 7-14, 58-61, 65-71, 78,79
A	paragraph '09.1!	6,15-57, 62-64, 72-77, 80-85

X	WENGER S ET AL: "H.26L OVER IP AND H.324 FRAMEWORK" ITU TELECOMMUNICATIONS STANDARDIZATION SECTOR VCEG-N52, XX, XX, 18 September 2001 (2001-09-18), pages 1-13, XP001148203 page 1 -page 3	1-5, 7-14, 58-61, 65-71, 78,79
A	page 6 -page 7	6,15-57, 62-64, 72-77, 80-85

A	WIEGAND T: "JOINT MODEL NUMBER 1, REVISION 1(JM-IRL)" ITU STUDY GROUP 16 - VIDEO CODING EXPERTS GROUP, XX, XX, 3 December 2001 (2001-12-03), pages 1,3-75, XP001086627 the whole document	1-85

A	D. SINGER, W. BELKNAP, G. FRANCESCHINI: "ISO Media File format specification" ISO/IEC JTC1/SC29/WG11 (N4270-1), 23 July 2001 (2001-07-23), XP002245422 the whole document	1-85

A	SINGER D ET AL: "AMENDMENT 6: MP4, THE MPEG-4 FILE FORMAT" ISO/IEC JTC1/SC29/WG11 MPEG01/N4420, XX, XX, 7 December 2001 (2001-12-07), pages I-II,1-10, XP001058448 the whole document	1-85